

Z MIKROKOMPUTEREM NA TY
8
NR INDEKSU
PL ISSN 0860-1674
Bajtek
MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"
NR 8
CENA 100 ZŁ
W NUMERZE
PORADNIK MŁODEGO PIRATA
CPC 6128
SZCZUR ROLAND W AKCJI
OCZY ZIEMI

CIĄG DALSZY

Mam kolegę — dziennikarza specjalizującego się w tematyce zagranicznej. Życie dziennikarskie, zwłaszcza gdy patrzy się na nie z boku, posiada wiele uroku, ale tak naprawdę przynosi również wiele stresów. Zamiast się bowiem cieszyć z szansy towarzyszenia wielkiej polityce, z wyjazdów zagranicznych, kolega coraz częściej narzeka. Kiedyś narzekał na magnetofon, który przydzielała mu na wyjazdy macierzysta redakcja. W czasie gdy koledzy z innych krajów dysponowali małymi, leciutkimi magnetofonikami reporterskimi „nasz człowiek" musi telepać się po świecie z pięciokilogramowym pudłem i trzema kompletami przedłużaczy i gniazdek ... Było, minęło — kolega wziął pożyczkę w Kasie Zapomogowo-Pożyczkowej, poszedł do komisu i na magnetofon już nie narzeka. Obecnie głównym tematem prowadzonych przez niego rozmów są mikrokomputery, a konkretnie ich brak.

Od kilku lat — opowiada kolega — na ważnych międzynarodowych imprezach obsługiwanych z reguły przez agencje i gazety z całego świata, coraz częściej pojawiają się faceci z małymi prostokątnymi walizeczkami zawierającymi przenośne mikrokomputery. Zamiast po imprezie biegać po biurze prasowym w poszukiwaniu wolnej maszyny do pisania, potem, po napisaniu relacji, pracowicie dodawać ogonki i kreski do różnych ę, ą, ó, ł itp., a potem stać w kolejce do teleksu albo przez 20 minut wykrzykiwać do słuchawki telefonicznej treść korespondencji — dziennikarz z walizeczką działa zupełnie inaczej. Jeszcze w trakcie imprezy rozkłada komputerek na kolanach i — na gorąco — wystukuje na klawiaturze relację. Nie musi przejmować się błędami i chaotycznością zapisu, gdyż załadowany do komputera program do redagowania tekstów robi wiele za niego. Poprawki, zamiana nazwisk, dopiski, wstawki, zmiana szyku, relacji — to dla komputera kwestia sekund.

Kilka minut po imprezie, przejrzawszy na płaskim ekranie całość relacji facet z walizeczką podchodzi do najbliższego telefonu i wykręca numer redakcji. Za pomocą specjalnej przystawki nakładanej na mikrotelefon i modemu czyli układu przekształcającego informację w postać akustyczną, sprzęga swój komputer z komputerem redakcji. Transmisja całej relacji zajmuje tą drogą kilkadziesiąt sekund. I po robocie! Dodajmy, że po ewentualnym przejrzeniu i „opisaniu" tekstu w redakcji, relacja taka przekazywana była zazwyczaj bezpośrednio do — także skomputeryzowanego — składu drukarskiego.

Kolega obliczył, że za służbowe pieniądze, jakie wydał w ubiegłym roku na opłacanie połączeń telefonicznych podczas obsługi wydarzeń międzynarodowych (spotkanie na szczycie w Genewie, konferencja 10-lecia KBWE w Helsinkach i kilka innych) można byłoby zakupić przenośny komputer z modemem, który w ciągu następnych lat pracowałby już „za darmo". Wrodzony optymizm każe nam obu wierzyć, że również i polscy dziennikarze otrzymają w niezadługiej przyszłości małe walizeczki pozwalające przede wszystkim... oszczędzać pieniądze!

Wspomniałem o tych problemach z zakresu dziennikarskiej „kuchni", gdyż są one ilustracją szerszego zjawiska. Otóż coraz częściej pisząc i mówiąc o roli i znaczeniu komputerów dla rozwoju cywilizacji, o sztucznej inteligencji, o rozpoczynaniu szerokiej edukacji informatycznej itp. jakby zapominamy o tym, że wprowadzanie tych urządzeń ma sens tylko wtedy, jeśli w sposób rzeczywisty ułatwiają one pracę ludzką. Wszystko inne też jest oczywiście ważne, ale jednak drugorzędne.

Tymczasem w Polsce większość mikrokomputerów ma na razie jedno tylko zastosowanie — zabawowe. Nie jesteśmy oczywiście w „Bajtku" przeciwni grom komputerowym, upieramy się jednak, że bardzo źle byłoby gdyby cała edukacja komputerowa miała służyć tej, nie aż tak ambitnej przecież sprawie.

Zabawa i rozrywka komputerowa jest oczywiście ważna, ale tylko jako wstęp do dalszego kontaktu z komputerem. I ten ciąg dalszy musi być dostępny nie tylko w książkach i filmach, ale młodzi adepci informatyki muszą go widzieć w normalnym życiu — tu i teraz. Brak tego ciągu dalszego może okazać się groźny w skutkach, m.in. dlatego, że pieniądze wykładane obecnie na szeroką edukację komputerową pójdą na marne. Byłaby to niewybaczalna strata tempa.

Na razie końcówki komputerowe można zobaczyć u nas w biurach LOT-u i... właściwie nigdzie więcej. Pomijam tu oczywiście zastosowania komputerów, nieliczne zresztą, w specjalistycznych sieciach informatycznych i w sterowaniu procesami produkcyjnymi. Tymczasem — starałem się to wykazać na przykładzie mego kolegi „międzynarodnika" — możliwości praktycznych zastosowań komputerów w naszym życiu społeczno-gospodarczym jest już obecnie tak dużo, że żadne usprawiedliwienia nie mogą być przyjęte. Zwłaszcza, że i argument bariery finansowej jest, jak się okazuje, bardzo łatwy do obalenia, bo i tak te pieniądze nawet bez komputerów wydajemy — i to nawet dużo więcej.

Coraz więcej ludzi w Polsce oswaja się z myślą, że szerokie zastosowanie komputerów jest w przyszłości rzeczywiście nieuchronne. Rzecz w przekonaniu ich, że ta nieuchronność zaczyna coraz bardziej dotykać również dnia dzisiejszego. Że stać już nas na to. I że nam wszystkim bardzo się to może opłacić!

Waldemar Siwiński



WYBIERZ SAM

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH".

ADRES: 00-687 Warszawa, ul. Wspólna 61. Telefon 21-12-05.

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu), Oskar Bramski, Krzysztof Czernek, Wiesław Migut, Sławomir Polak, Roman Poznański, Wanda Roszkowska (opr. graficzne), Marcin Waligórski, Roman Wojciechowski.

Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.

Cena 100 zł.

Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.

Zam. nr 1051/86, nakład 250.000 egz. P-101



Bajtek



Rozmowa z Tomaszem Pyciem, dziennikarzem telewizyjnym, autorem programu „Spektrum"

— Czytelnicy „Bajtka" chętnie oglądają „Spektrum"....

— ...nie zawsze zapewne wiedząc, iż nazwa programu nie ma nic wspólnego z popularnym komputerem „Spectrum". Nasze „Spektrum" ma w środku „k" i oznacza widmo światła białego, jest to więc pojęcie znane już z lekcji fizyki...

— Ale nie od początku mówiliście w „Spektrum" o komputerach..

— Rzeczywiście, nie od samego początku. Program powstał w roku 1984, a problematyka komputerowa „dojrzała" u nas dopiero w roku następnym. Warto od razu wyjaśnić zasadniczą różnicę między tym programem a „Sondą". Wymyśliliśmy „Spektrum" wspólnie z **Bogdanem Borusławskim** jako magazyn aktualności, najświeższych nowinek technicznych, a więc jako zestaw wiadomości, które w monotematycznych „Sondach" nie można było prędko zmieś[illegible]

— Młody człowiek oglądający was na ekranie myśli pewnie z zazdrością o dziennikarzach telewizyjnych.. Zamawiają materiały na całym świecie, sprowadzają sobie sprzęt...

— [illegible] jeszcze?! Wiem [illegible] rzy tak sobie wyobrażają naszą pracę. Rzeczywistość jest jednak znacznie mniej kolorowa. Materiały zdobywamy drogą wymiany z innymi telewizjami i organizacjami, a także dzięki własnym kontaktom, których oczywiście nie zdradzę... A jeśli chodzi o sprzęt... No, cóż, kiedy **Andrzej Kurek** przyniósł do studia własne „Spectrum 48 K" zastanawialiśmy się jak to urządzenie wykorzystać praktycznie. Zrobiliśmy wówczas czołówkę programu, która zresztą ukazuje się do tej pory.

— No właśnie, dziś nie jest to już chyba w telewizji żadna rewelacja....

— Rewelacja to nie jest, ale w większości programów cały czas plansze tytułowe i końcowe są przekładane ręką jakiejś miłej pani. My używamy komputerów do napisów i grafiki chociażby dlatego, że rezygnacja z usług plastyków znacznie obniża koszty programu. Ale używamy komputerów prywatnych lub pożyczonych od życzliwych ludzi. Telewizję mamy jaką mamy, nie stać jej na jeden choćby komputer graficzny. Są tylko syntetyzery [illegible] do na[illegible] średniej zresztą jakości.

— To znaczy, że nie macie w studiu ani jednego służbowego komputera?

— Ani jednego. Czołówkę muzyczną **Marek Biliński** też nagrał na swoim sprzęcie.

— Bez kłopotu można pożyczyć drogie komputery?

— Często im droższe, tym łatwiej pożyczyć. A tak na marginesie mogę wam powiedzieć, że kiedyś realizowaliśmy program o robotach. Bez trudu pożyczyliśmy od jednej z firm dwutonowego robota przemysłowego, natomiast poważne kłopoty sprawiło nam pożyczenie na kilka dni małej zabawki z Muzeum Techniki. Była to makieta robota z migającą na górze żarówką. Do wypełnienia wszystkich pismek zaangażowano w muzeum prawie 10 osób.

— Całe szczęście, że widzowie nie muszą o tym wszystkim wiedzieć. Czy orientujecie się jednak, jacy są wasi odbiorcy? Ile mają lat, jakie wykształcenie?

— W popularyzacji nauki i techniki trudno jest precyzyjnie określać krąg odbiorców. W przypadku technik komputerowych zdarza się, że podobne wiadomości mają profesor uniwersytetu i jego wnuczek. Na pewno więc nie ma znaczenia kryterium wieku. Nie można też ograniczać się do określonego stopnia trudności, bo przecież jedni chcą fachowych szczegółów technicznych, a drudzy zwykłych ciekawostek. Musimy pamiętać o wszystkich. Dlatego między innymi pokazaliśmy i omówiliśmy praktycznie, prawie wszystkie dostępne w tej chwili w kraju komputery.

Jest natomiast jeden element wspólny dla wszystkich odbiorców: komputery budzą emocje. Widać to szczególnie dobrze, gdy ogłaszamy konkursy. Przychodzą wówczas worki listów. A jeśli jeszcze główną nagrodą jest mikrokomputer, to już u niektórych wywołuje gorączkę...

— Nie spotykasz się z zarzutem, że pokazujecie coś ciągle niedostępnego, elitarnego...

— Takie zarzuty stawiane są rzadko. Ale zawsze wówczas pytam, czy lepiej jest nie pokazywać, nie mówić, udawać, że nie ma? Mnie nie interesuje, czy pokazywany sprzęt jest za złotówki czy za dolary i ile kosztuje. Ważne jest to, że istnieje. To trochę tak jak z Eskimosami, których większość z nas nie oglądała, ale to jeszcze nie powód, żeby o nich nie wspominac. A z komputerami jest o tyle lepiej, że są dostępne w klubach i u niektórych kolegów. I ciągle ich przybywa! Zresztą po to między innymi jest telewizja, żeby pokazywać to, co niedostępne. To zresztą najbardziej pociąga...

— Gdybyście pokazywali tylko to, co jest łatwo osiągalne, to rzeczywiście nie byłyby wasze programy specjalnie ciekawe...

— Telewizja przekazuje informacje, ale i pobudza wyobraźnię, wzbudza zainteresowanie. Uważam zresztą, że samo czytanie fachowych książek i pism w przypadku komputerów daje niewiele. Wiem to z własnego doświadczenia. Więcej można się nauczyć oglądając choćby programy telewizyjne, a już najwięcej, rzecz jasna, mając dostęp do klawiatury. Niekoniecznie od razu w domu, wystarczy w klubie komputerowym. I wtedy dopiero okazuje się, jak potrzebna jest fachowa prasa i literatura... W każdym razie ja taką proponuję kolejność.

— Pewnie nie ma tu jednej zasady, wasi widzowie też mogliby mieć różne opinie. Dużo dostajecie listów?

— Sporo. Od skazanych z góry na niepowodzenie próśb o kopiowanie gier, do listów z prośbami o konkretne porady. Niektórzy mają nam za złe, że pokazujemy zabawowe funkcje komputerów, ale uważam, że gry komputerowe to zabawa, z której się po prostu wyrasta. Ja też dopiero po jakimś czasie użytkowania komputera zorientowałem się, że mam w ręku coś naprawdę cennego, cenniejszego niż początkowo sądziłem.

Rozmawiali:
Sławomir Polak
Piotr Radziszewski

RENTA — Z KOMPUTERA —



Tysiące urzędniczek, studenci i młodzież szkół średnich, od świtu do zmroku, od wiosny aż do jesieni pochyleni nad kolumnami cyfr. Tak było do niedawna. Dzisiaj zastosowanie komputerów umożliwia Zakładowi Ubezpieczeń Społecznych obliczenie podwyżki rent i emerytur dla pięciu milionów osób w niespełna dwa miesiące.

Zanim na serio zaczęto myśleć o komputerowym systemie obliczania i wypłacania rent oraz emerytur wydziały i oddziały ZUS zbierały cięgi za opieszałość.

Wdrożenie komputerowego systemu obliczania danych nie od razu przyniosło spodziewane efekty. Zawiniła m in. struktura samej instytucji. Pięćdziesiąt pięć oddziałów terenowych robiło swoje, centrala w Warszawie usiłowała zgarnąć bractwo pod swoje skrzydła. W kilkudziesięciu oddziałach obowiązywał jeden z trzech programów obliczania wysokości świadczeń. Programy do dzisiaj są różne, ponieważ są dziełem tych, którzy oddziałom ZUS udostępniają swoje bazy komputerowe. Zgranie pracy 30 przedsiębiorstw Zakładu Elektronicznej Techniki Obliczeniowej z postępami we wprowadzaniu nowych danych w 55 oddziałach ZUS, posługujących się trzema programami, jeszcze dzisiaj bardziej utrudnia, niż ułatwia przeprowadzenie akcji rewaloryzacji lub podwyższania rent i emerytur. Ale przed komputerami ZUS nie jest już w stanie uciec.

Powiało nadzieją

Początki tworzenia skomputeryzowanego systemu ZUS sięgają 1978 roku. Od tamtej pory, jak twierdzą informatycy z koordynującego pracę stołecznego Centrum Informatyki Ubezpieczeniowej, wielu zrobiło wiele. Narodziły się programy komputerowe:: EMIR, MIREN i RENTIER, ale rozmnożyły się również ich mutacje: kielecka, koszalińska, warszawska i dwie bydgoskie. Jednak dopiero w b.r. przeprowadzono generalne porządki. Cztery oddziały ZUS pracują w systemie RENTIER /obliczenia wykonuje katowicka filia ZETO, dwanaście ośrodków poddało się systemowi EMIR (bygdoskie ZETO), pozostałe 39 oddziałów ZUS przekonało się do nowego systemu WARENT.

Do 15 lipca musieliśmy wykonać oprogramowanie dla wykonania tegorocznej podwyżki rent i emerytur — tłumaczy mgr Marian Kacperski, wicedyrektor CIU ZUS. Mając do dyspozycji komputery: RIAD 32 oraz IBM 43 31 mogliśmy stworzyć własny program na użytek najpierw dwóch warszawskich oddziałów ZUS. Tak powstał WARENT, program pozwalający komputerowi wyszukiwać odpowiednie dane, mnożyć je przez dwa współczynniki i przekazywać drukarce do wypisania na kwicie dla odbiorcy. Dzięki temu wyeliminowaliśmy błędy, które powstają podczas ręcznego wypełniania formularzy, komputer nie pomylił się podczas mnożenia. Podwyżka trwa nie sześć miesięcy a sześć tygodni.

WARENT stworzono tylko dla wykonania tegorocznej podwyżki, nie można dzięki niemu wykonywać skomplikowanych obliczeń. Dlatego część oddziałów nadal pracuje w systemie EMIR lub RENTIER, które pozwalają obliczać wszystkie sytuacje zmiany danych, związane z rentami emeryturami. Niestety żaden z tych systemów nie pozwala w ogóle wyeliminować ręcznych obliczeń, wyszukiwania teczek w szafach, nanoszenia zmian na specjalne karty. Systemy te „przyjmują" zmiany danych, ale w zamian „gubią" dane starsze, które mogą być potrzebne w innej sytuacji.

Zamiast szaf

Na zapamiętywanie danych, porównywanie zmiennych, wyszukiwanie w magnetycznej pamięci nowych i starych aktów prawnych, decyzji różnych urzędów może sobie pozwolić komputer zaprogramowany w systemie REM. Znikają szafy z dokumentami, nie potrzeba danych z pismami i zaświadczeniami, wzrasta nieomylność, kurczy się czas operacji obliczeniowych. System ten wymaga jednak komputerów dużej mocy. Stąd też próbnie zastosowano go do obsługi grupy emerytów i rencistów odbierających świadczenia w jednym terminie wysłane przez oddział ZUS w Siedlcach. Ośrodek ten, jako jedyny w kraju, uzyskał stałe łącze z komputerem IBM w Warszawskim Centrum Informatyki Ubezpieczeniowej.

Wszyscy zgadzają się, że najpełniejszym systemem byłby REM, ale sieć placówek ZETO dysponuje komputerami RIAD i MERA, które nie podołają takiej lawinie danych — mówi Krzysztof Lisicki, kierownik wydziału informatyki I Oddziału ZUS w Warszawie. Rozpowszechniamy więc WARENT, którego koncepcja narodziła się w CIU ZUS, a oprogramowanie powstało w katowickim ZETO. Staramy się objąć tym systemem oba stołeczne oddziały, Siedlce i większość podwarszawskich placówek ZUS. Jesteśmy jednak ograniczeni skąpymi przydziałami łączy do teletransmisji.

Komputer i dziewczyny

W stołecznym I Oddziale ZUS jest już zainstalowane stałe łącze, są wydzielone pomieszczenia do ustawienia monitorów współpracujących z IBM w Centrum Informacji Ubezpieczeniowej. Przygotowania zostały spowodowane planami przechodzenia na system REM. Przygotowano już prawie wszystko — najdłużej trwa remont samych pomieszczeń.

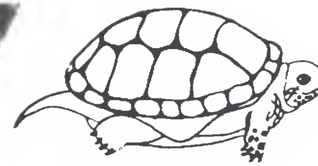
Tymczasem pokoje na parterze okupuje czternaście ślicznych dziewcząt. Tworzą tzw. wydział uzupełniania danych na zdjęciu. Z kartonowych formularzy odczytują datę urodzenia i płeć emeryta lub rencisty oraz rok przyznania świadczenia. Jednocześnie, nie patrząc na klawiaturę, wystukują dane, które natychmiast pojawiają się na niebieskim kineskopie monitora.

Dziennie „wystukuję" ok. 21 600 znaków, ale to wcale nie jest męczące — uśmiecha się Mariola Motak. Trafiłam tutaj przypadkowo, po skończeniu Technikum Mechaniczno-Precyzyjnego. Po zastosowaniu teletransmisji — mówi Sławomir Załęski, naczelnik wydziału wprowadzania danych, nie trzeba będzie wozić taśm magnetycznych do centrali z IBM. A jeśli zostanie wdrożony system REM, wszelkie zmiany będą przeprowadzone w łączności z komputerem. Po wywołaniu informacji, naniesieniu zmian i sprawdzeniu, dane zostaną automatycznie wprowadzone do pamięci komputera.

Na razie takie obrazki to jeszcze przyszłość. ZUS raczej nie może liczyć na przydziały dużych profesjonalnych mikrokomputerów z pamięcią o pojemności rzędu 100 megabajtów. Być może urządzenia takie trafią do filii ZETO. Wtedy dopiero w ZUS znikną szafy wypchane papierami, a dwóch urzędników będzie mogło wykonywać dzisiejszą pracę 20 osób. Podstawowe formularze lub dyskietki nie będą „podwożone" samochodami do komputerów.

Pozostaje więc nam czekać, udoskonalać to, co już mamy i nie zapominać, co chcemy mieć.

Wojciech Gładykowski



LOGO

SŁOWNIK MINIMUM cz. IV

Oto czwarta i ostatnia część mini–słownika LOGO. Przekazaliśmy niniejszym czytelnikowi kompletny opis dialektów Sinclair LOGO i Terrapin LOGO, a w dużej części — Dr LOGO, Commodore LOGO i Atari LOGO. Mamy nadzieję, że okaże się on pomocny podczas własnych poszukiwań.

4.1. Procedury robocze

ERALL

Usunięcie z pamięci roboczej komputera wszystkich aktualnie istniejących procedur i zmiennych.

```
TO PROCEDURA
PRINT :zmienna
END
MAKE "zmienna "wartosc
ERALL
PROCEDURA
I don't know how to PROCEDURA
PRINT :zmienna
zmienna has no value
```

ERASE s1 ER s1

Usunięcie z pamięci roboczej komputera procedury lub zmiennej, o nazwie podanej jako parametr (ew. procedur o nazwach zawartych w liście).

```
TO PROCEDURA
PRINT :zmienna
END
MAKE "zmienna "wartosc
ERASE "procedura
PROCEDURA
I don't know how to PROCEDURA
```

Usunięcie z pamięci roboczej komputera procedury lub zmiennej o nazwie podanej jako parametr. ERASE może byc również użyte w jeden z następujących sposobów:

ERASE PROCEDURES
— usuwa z pamięci wszystkie aktualnie istniejące procedury;

ERASE NAMES
— usuwa z pamięci wszystkie aktualnie istniejące zmienne;

ERASE ALL
— stanowi złożenie ER PROCEDURES I ER NAMES.

ERN s1

Usunięcie z pamięci roboczej komputera wymienionej zmiennej (lub zmiennych).

```
MAKE "zmiennal 123
MAKE "zmienna2 "halo
MAKE "zmienna3 "zmiennal
ERN [zmiennal zmienna2]
PRINT THING :zmienna3
zmiennal has no value
```

Usunięcie z pamięci roboczej komputera wymienionej zmiennej.

```
MAKE "alfa 5
ERNAME "alfa
PRINT :alfa
alfa has no value
```

ERNS

Usunięcie z pamięci roboczej wszystkich aktualnie istniejących zmiennych.

```
TO NAGRAJ.PROCEDURY :plik
ERNS
SAVEALL :plik
END
NAGRAJ.PROCEDURY "grafika
```

ERPS

Usunięcie z pamięci roboczej komputera wszystkich aktualnie istniejących procedur.

```
TO PROCEDURA
PRINT :zmienna
END
MAKE "zmienna "wartosc
ERPS
PROCEDURA
I don't know how to PROCEDURA
PRINT :zmienna
wartosc
```

PO s1

Wydrukowanie treści wymienionej procedury (procedur).

```
TO PROCEDURA.PUSTA
END
PO "procedura.pusta
TO PROCEDURA.PUSTA
END
```

POALL

Wydruk treści wszystkich aktualnie istniejących procedur oraz zmiennych wraz z ich wartościami.

```
POALL
```

PONS

Wydruk nazw i wartości wszystkich aktualnie istniejących zmiennych.

```
PONS
```

POPS

Wydruk treści wszystkich aktualnie istniejących procedur.

```
POPS
```

Wydruk nazw i nazw parametrów wszystkich aktualnie istniejących procedur.

```
POTS
TO KULA :promien
TO ELIPSOIDA :a :b :c
TO WIELOSCIAN :ilescian :krawedz
```

Wydrukowanie treści wymienionej procedury. PRINTOUT może być użyte również w jeden z następujących sposobów:

PRINTOUT PROCEDURES
— wydruk treści wszystkich aktualnie istniejących procedur;

PRINTOUT NAMES
— wydruk nazw i wartości wszystkich aktualnie istniejących zmiennych;

PRINTOUT ALL
— stanowi złożenie PO PROCEDURES I PO NAMES

PRINTOUT TITLES
— działa jak POTS

4.2. Pamięć zewnętrzna

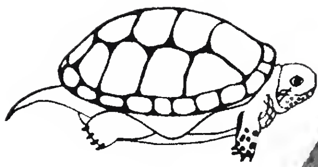
CATALOG

Wydrukowanie nazw wszystkich znajdujących się na dysku (kasecie Microdrive'u) plików. Nie działa dla pamięci kasetowej. W dialekcie Dr LOGO podobny efekt uzyskać można przez PR DIR.

```
CATALOG
```

Usunięcie z dysku (lub kasety Microdrive'u) pliku o podanej nazwie. Nie działa dla pamięci kasetowej.





```
ERASEFILE "grafika
```

[illegible]

Usunięcie z dysku pliku o podanej nazwie, uprzednio utworzonego przez SAVEPICT (zawierającego grafikę).

LOAD s

Załadowanie do pamięci roboczej wskazanego pliku (z magnetofonu, dysku lub Microdrive'u).

```
LOAD "grafika
```

LOADD s

Załadowanie z urządzenia pamięci zewnętrznej wskazanego pliku, stanowiącego zawartość edytora.

```
LOADD "list
```

LOADSCR s

Załadowanie z urządzenia pamięci zewnętrznej wskazanego pliku, stanowiącego zawartość ekranu.

```
LOADSCR "wzorek
```

[illegible]

Patrz LOAD

READPICT s

Patrz LOADSCR

SAVE s s|

Nagranie wymienionej w parametrze drugim procedury (procedur) na plik o nazwie określonej przez pierwszy parametr.

```
SAVE "figury [kwadrat trojkat
wielokat kolo]
```

SAVE s

Patrz SAVEALL

SAVEALL s

Nagranie aktualnej zawartości pamięci roboczej (procedur i zmiennych) na plik o podanej nazwie.

```
SAVEALL "grafika
```

SAVED s

Nagranie aktualnej zawartości edytora na plik o podanej nazwie.

```
SAVED "list
```

SAVEPICT s

Patrz SAVESCR

SAVESCR s

Nagranie aktualnej zawartości ekranu graficznego na plik o podanej nazwie.

```
SAVESCR "obrazek
```

SETDRIVE n

Zmiana aktualnie używanego urządzenia pamięci zewnętrznej. Dla Spectrum parametr musi mieścić się w przedziale 0..8, gdzie 0 określa magnetofon, natomiast liczby 1..8 odpowiadają kolejnym Microdrive'om.

```
SETDRIVE 1
CATALOG
```

4.3 Drukarka

COPYSCREEN

Skopiowanie zawartości ekranu (w trybie graficznym lub tekstowym) na drukarce.

```
COPYSCREEN
```

PRINTON

Włączenie opcji, przy której wszystkie teksty wyprowadzane na ekran są równocześnie drukowane. W Dr LOGO efekt ten można uzyskać przez COPYON.

```
PRINTON
PRINT [Dzien dobry !]
```

PRINTOFF

Powrót do trybu, w którym teksty wyprowadzane są tylko na ekran monitora. (W Dr LOGO — COPYOFF)

```
PRINTON PRINT "Alfa
Alfa
PRINTOFF PRINT "Beta
Beta
```

4.4. Pamięć

NODES → n

Aktualna ilość wolnej pamięci roboczej, mierzona w węzłach (1 węzeł = 5 bajtów).

```
ERALL RECYCLE PRINT NODES
2288
```

RECYCLE

Procedura zwalniająca maksymalną ilość pamięci roboczej poprzez jej uporządkowanie. W toku pracy komputera porządkowanie takie jest wykonywane automatycznie zawsze, ilekroć zabraknie pamięci do wykonywania obliczeń. Uprzednie użycie RECYCLE pozwala uniknąć nieoczekiwanych opóźnień w wykonywaniu programu.

```
RECYCLE
```

.BLOAD s n

Załadowanie z urządzenia pamięci zewnętrznej pliku utworzonego przez — BSAVE) o nazwie określonej przez pierwszy parametr i umieszczenie go w pamięci począwszy od adresu równego podanej liczbie.

```
.BLOAD "obraz 64284
```

.BSAVE s 1

Nagranie na plik o podanej nazwie pewnej ilości bajtów pamięci, określonej przez drugi parametr. Jest on listą dwuelementową, której pierwszy element określa adres początkowy, a drugi — długość danego odcinka pamięci.

```
.BSAVE "obraz [64284 200]
```

.CALL n

Wywołanie procedury w języku zewnętrznym, rozpoczynającej się pod podanym adresem pamięci.

```
TO ZNISZCZ.LOGO
.CALL 0
END
ZNISZCZ.LOGO
```

.DEPOSIT n n

Umieszczenie pod adresem określonym przez parametr pierwszy bajtu stanowiącego parametr drugi.

```
TO UMIESC :adres :lista
IF EMPTYP :lista [STOP]
.DEPOSIT :adres FIRST :lista
UMIESC :adres + 1 BF :lista
END
UMIESC 65368 [85 0 85 0 85 0
85 0]
PRINT CHAR 144
```

.EXAMINE n → n

Wartość wskazanej komórki pamięci.

```
PRINT .EXAMINE 0
243
```

.PRIMITIVES

Wydruk nazw wszystkich procedur pierwotnych LOGO.

```
.PRIMITIVES
```

.RESERVE n

Zarezerwowanie wskazanej ilości bajtów pamięci roboczej na programy w języku wewnętrznym. W Spectrum rezerwowana przestrzeń zajmuje miejsce poniżej adresu 65024. .RESERVE zmniejsza ilość dostępnej pamięci roboczej.

```
.RESERVE 100
PRINT .RESERVED
64924 65024
```

.RESERVED → l

Lista, złożona z początkowego i końcowego adresu przestrzeni, zarezerwowanej uprzednio przez .RESERVE.

```
.RESERVE 200
PRINT .RESERVED
64824 65024
```

4.5. Inne

BYE

Zakończenie pracy w LOGO.

```
BYE
```

WAIT n

Przerwanie realizacji programu przez wskazaną ilość jednostek czasu, równych w przybliżeniu 1/50 sekundy.

```
TO PISZCZ
WAIT 45
SOUND [0.1 24]
PISZCZ
END
PISZCZ
```

Marcin Waligórski

NIE BÓJ SIĘ MNIE!

MÓJ EKRAN

Czy siedząc przed ekranem mojego monitora zastanawiałeś się kiedyś nad tym, że jeszcze niedawno, rozmowa z komputerem nie była tak wygodna jak dzisiaj? Moi dziadkowie, komputery sprzed kilku i kilkunastu lat mogliby jedynie pisać do ciebie listy na drukarkach, urządzeniach wówczas wielkich, bardzo zawodnych i niesłychanie hałaśliwych (ta ostatnia niedogodność do dzisiaj nie jest w pełni opanowana).

Tony papieru i kilometry taśmy barwiącej potrzebne były więc do pracy mojemu dziadkowi. Jeśli jeszcze dodać do tego fakt, że czytelność wydruków pozostawiała wiele do życzenia, to obraz pracy człowieka i komputera parę lat temu nie przedstawiał się różowo.

Tak się jednak składa, że bardzo szybko przyzwyczajamy się do rzeczy dobrych i trudno nam sobie wyobrazić, że kiedyś można się było bez tego obejść. Dzisiaj praktycznie każdy mikrokomputer — i ten duży i ten malutki — potrafi drukować wszystko na monitorze. Co więcej, ja i moi koledzy — komputery domowe — możemy współpracować ze zwykłymi telewizorami. Z jednym wszakże zastrzeżeniem: większość z nas jest przystosowana do współpracy z kolorowymi telewizorami pracującymi w systemie PAL, a więc jeśli posiadasz odbiornik czarno-biały lub kolorowy w systemie SECAM niestety będziesz oglądał obraz czarno-biały.

Przy okazji chciałem ci zwrócić uwagę na jeden fakt, z którego nie wszyscy zdają sobie sprawę. W większości przypadków, współpracując z odbiornikiem telewizyjnym jestem podłączany do gniazda antenowego. Nie jest to wyjście najbardziej szczęśliwe, a wynika jedynie z faktu, że większość telewizorów po prostu nie ma możliwości innego podłączenia. Jeśli jednak twój telewizor posiada tak zwane wejście monitorowe, znacznie lepiej zrobisz podłączając mnie bezpośrednio do tego wejścia. Unikniemy w ten sposób podwójnego transformowania sygnału: u mnie i w telewizorze.

My, komputery domowe zostałyśmy przygotowane do współpracy z odbiornikami TV tylko dlatego, że telewizor jest w każdym domu i w związku z tym odpada dodatkowy zakup. Jest jednak kilka argumentów przemawiających za tym, że jednak warto narazić się na dodatkowe koszty — nie tak znowu duże w porównaniu z moją ceną — i kupić monitor. Pierwszy z tych argumentów jest oczywisty i nasuwa się sam: w ten sposób unikniesz awantur w domu o to czy włączasz komputer, czy też oglądasz wraz z rodziną panią Stefanie Harper w towarzystwie kolejnego krokodyla.

Są jednak i inne, a to mianowicie jakość obrazu. Słyszałeś zapewne pojęcie rozdzielczość ekranu. Podaje się ją w punktach. I tak na przykład mój przyjaciel, Commodore 64 posiada rozdzielczość ekranu 320 × 200 punktów. To samo pojęcie odnosi się także do telewizorów i monitorów. Dobrze jest, gdy rozdzielczość monitora jest znacznie większa niż komputera gdyż gwarantuje to dobrą czytelność obrazu. Zupełna tragedia jest wówczas, gdy monitor posiada niższą rozdzielczość niż komputer. Na szczęście w moim przypadku użycie telewizora daje zupełnie przyzwoite wyniki (choć oczywiście mogłoby być lepiej).

Jeśli zdecydujesz się na zakup monitora staniesz przed koniecznością wyboru: zdecydować się na monitor kolorowy czy monochromatyczny (jednobarwny). Tym razem sam musisz odpowiedzieć sobie na pytanie, w jaki sposób będziesz korzystał z moich usług. Jeśli chcesz w głównej mierze pracować na tekstach i pisać programy, najlepszy dla ciebie będzie monitor z ekranem w kolorze zielonym lub miodowym. Ma on tę nieocenioną zaletę że znacznie mniej męczy oczy. Posiada także większą rozdzielczość.



Jeśli jednak twoją największą pasją są gry komputerowe, lub też będziesz korzystał z programów graficznych, musisz zdecydować się na monitor kolorowy.

Wspomniałem tutaj o bardzo ważnej rzeczy. Długie siedzenie przed ekranem nie jest specjalnie zdrowe dla Twoich oczu. I chociaż bardzo lubię gdy pracujesz i bawisz się ze mną, muszę cię przed tym przestrzec. Ponieważ jednak jestem przekonany, że moja życzliwa rada nie wpłynie w najmniejszym stopniu na ilość godzin spędzanych przez ciebie przy klawiaturze, pozwól, że udzielę ci przynajmniej kilku rad:

— Siadaj zawsze jak najdalej od ekranu, jednak w takiej odległości, aby napisy były dla ciebie bez trudu czytelne.

— Bardzo starannie dostrajaj telewizor. Unikaj migotania obrazu i staraj się by był on zawsze „ostry".

— Jeśli nie jest ci potrzebny obraz kolorowy, wyłączaj kolor.

— Jaskrawość obrazu ustawiaj na możliwie niskim poziomie, takim jednak, by napisy były wyraźne.

Mam nadzieję, że po przeczytaniu moich rad odsuniesz się nieco wraz z klawiaturą od telewizora. Jeżli nie, to obawiam się, że za kilka lat nie będzie można zobaczyć na ulicy młodego człowieka bez okularów.

Czego nie życzy ci w żadnym razie

Twój Komputer

DLA KAŻDEGO

kiedy 2=10

W książce Czesława Kosniowskiego („Maths on CBM-64") znalazłem następujący algorytm określający przeliczenie liczby dziesiątkowej N na dany układ liczenia B (tzn. dwójkowy, trójkowy itp.).

Mówi on, iż należy

1. Obliczyć M = INT (N/B)
2. Obliczyć R = N-B-M czyli resztą z dzielenia N/B
3. Podstawić M za N czyli N = M
4. Powtarzać 1, 2, 3 do momentu aż M = 0.

Kolejno otrzymane reszty R_1, R_2, R_3 ... R_s to cyfry s-cyfrowej liczby w układzie B.

Otrzymujemy zatem liczbę (N w układzie B): R_sR_{s-1} ... R_3 R_2 R_1 R_0

Program oblicza postacie liczb w układach od dwójkowego do 35-kowego zastępując (w przypadku systemów o podstawie większej od 10) liczby 10, 11, 12 itp. literami A, B, C itd.

Jacek Niewęgłowski

```
1 ' Program przelicza liczby
2 ' w ukladach w zakresie od
3 ' binarnego do 35-kowego.

10 INPUT "Uklad danej liczby";s
20 INPUT "Dana liczba";x$
30 INPUT "Na jaki uklad przelic
zyc";b
35 ' Przeliczenie na uklad
36 ' dziesietny

40 FOR i=1 TO LEN(x$)
50 p=LEN(x$)-i
60 x=ASC(MID$(x$,i,1))
70 IF x>64 THEN x=x-55: GOTO 90
80 x=x-48
90 n=n+x*s^p: NEXT i
95 ' Przeliczenie na uklad B

100 m=INT(n/b):r=n-b*m:n=m
110 IF r>9 THEN a$=CHR$(r+55)+a
$: GOTO 130
120 a$=CHR$(r+48)+a$
130 IF n<>0 THEN 100
140 PRINT x$; " W ukladzie";b;"
-kowym:";a$
```

Wieczny kalendarz

Czy wiesz w jakim dniu tygodnia się urodziłeś? A może interesuje cię czy bitwa pod Grunwaldem miała miejsce w poniedziałek czy w piątek? Jeśli tak, wpisz do swojego komputera ten program i dowiesz się wszystkiego.

F. Głowiński

```
10 FOR i=0 TO 6
20 READ x$(i)
30 NEXT i
40 INPUT "rok,miesiac,dzien";r,m,d
50 IF m<3 THEN 70
60 GOTO 90
70 r=r-1
80 m=m+12
90 d=r+INT(r/4)-INT(r/100)+INT(r/400)+3*
m-INT((m+m+1)/5)+d+1
100 d=d-INT(d/7)*7
110 PRINT x$(d)
120 DATA niedziela, poniedzialek, wtorek
, sroda, czwartek, piatek, sobota
```

CPC

W niespełna rok po wejściu na rynek mikrokomputera CPC 464 i pół roku po debiucie „przejściowego" modelu (o którym parę słów pod koniec artykułu) CPC 664, firma Amstrad stała się nieomal monopolistą w kręgu wytwórców komputerów ośmiobitowych, wypuszczając CPC 6128.

Na uwagę zasługuje fakt, że w kolejno produkowanych modelach zasadnicza koncepcja konstrukcyjna pozostała niezmieniona, tzn. firma opierała się ciągle na kilku podstawowych założeniach: był to wciąż komputer z bardzo dobrze znanym i popularnym mikroprocesorem Z-80 a jego możliwości zostały wykorzystane prawie do maksimum: nabywca w każdym przypadku stawał się posiadaczem kompletnego minisystemu komputerowego (minikomputer, monitor i pamięć magnetyczna) przy relatywnie niskiej cenie.

Pomijając pewne zabiegi kosmetyczne, tzn. nieco mniejsze gabaryty, zmienioną kolorystykę (czarna obudowa, szare klawisze) i bardziej zwartą klawiaturę, Amstrad wprowadził w **CPC 6128** również kilka bardziej istotnych zmian, pozwalających zakwalifikować ten model do grupy minikomputerów „półprofesjonalnych". Zadecydowały o tym dwa elementy: powiększenie pamięci RAM do 128 kB i wbudowany na stałe zamiast magnetofonu (w 464) napęd dysków 3-calowych wraz z odpowiednim oprogramowaniem (**AMSDOS i CP/M** +).

Nadal jednak istnieje w tym modelu możliwość korzystania ze zwykłego, tyle, że już zewnętrznego, magnetofonu kasetowego. W stosunku do CPC 464, opisanego w „Bajtku" nr 5–6, model ten różni się jeszcze:

a) wielkością pamięci ROM — powiększoną do 48 kB, co jest konsekwencją zainstalowania napędu dysków. Pamięć została wzbogacona o system operacyjny dysku **(AMSDOS)**
b) nieco inną wersją języka Basic — tej samej firmy Locomotive Software lecz **BASIC 1.1.** Wersja ta różni się w zasadzie dodatkowymi sześcioma komendami obsługującymi grafikę komputera,
c) wbudowanym sprzęgiem (interfejsem) obsługującym wewnętrzny napęd dysków DD1 oraz dodatkowy zewnętrzny drugi napęd FD1,
d) dodatkowym zasilaczem + 12V w monitorze i dodatkowym przewodem doprowadzającym to napięcie do napędu dysków,
e) dołączanym i przy zakupie systemu dwoma dyskietkami, zawierającymi program demonstracyjny,. **CP/M**+ wraz z **DR.LOGO** i **CP/M 2.2** z **DR LOGO2.** cp/m 2.2 i dr.logo2 dołączane są wyłącznie w celu zapewnienia możliwości wykorzystywania programów pracujących w tym systemie na CPC 464 i 664.

Wyjaśnienia wymaga „zagadkowość" istnienia 128 kB pamięci w komputerze 8-bitowym, który posiadając 16-bitową szynę adresową może zaadresować tylko 64 k. Zastosowano tutaj pewien trick, dzieląc pamięć na dwa bloki po 64 kB pracujące na zmianę. Bloki takie nazwano bankami a przełączeniem ich steruje program firmowy znajdujący się na pierwszej stronie dyskietki. Program ten, ładowany z **BASIC**-a pod nazwą Bank Manager, powoduje uaktywnienie rozszerzenia systemowego (RSX) umożliwiając wykorzystywanie obu banków. Istnieje jednak pewne ograniczenie: drugie 64 kB pamięci nie może być wykorzystane dla pisania programów w **BASIC-u**. Można jednak w tym obszarze przechowywać dane dla tych programów, czyli traktować ten bank jako dysk typu **RAM**. Inną z możliwości wykorzystania drugiego banku jest chwilowe przechowywanie treści obrazów ekranu monitora. Ponieważ każdy bank podzielony jest na cztery sekcje po 16kB, więc można jednocześnie przechować 4 obrazy, wykorzystywane np. do zmiany scenerii gier komputerowych.

Dostarczany razem z komputerem **CP/M**+ nie stwarza żadnych problemów w ten sposób, że rezydując w zasadzie w jednym banku, pozostawia użytkownikowi na jego program ok. 61 kB. Jest to ulepszona w stosunku do **CP/M 2.2** wersja systemu operacyjnego odznaczajaca się m.in. większą szybkością dostępu do danych na dysku, możliwością rejestracji w katalogu dysku dat wykonywanych zapisów i odszukiwania takich zbiorów z określonego przedziału czasowego, prostszą pracą z dyskami i związaną z tym mniejszą szansą omyłkowego zniszczenie tworzonych i istniejących pro-



6128



gramów. Jednak największą zaletą **CP/M+** w **CPC 6128** jest znacznie większy obszar pamięci, będący w dyspozycji użytkownika (w **CP/M 2.2** tylko 39 kB) niż w poprzednich modelach 464 i 664. Pozwala to w pełni wykorzystywać istniejącą, olbrzymią bibliotekę profesjonalnych programów użytkowych pracujących pod tym systemem (np. Wordstar, dBase II czy Turbo Pascal).

Ze względu na pewną pozycję rynkową Amstrada, jak również fakt, że wykorzystuje w swoich komputerach stosowany i w innych firmach mikroprocesor Z80 (np. Spectrum czy rodzina minikomputerów MSX) do wszystkich jego modeli istnieje już mnóstwo programów nieprofesjonalnych (gry komputerowe, programy graficzne i edukacyjne). Pozycja rynkowa powoduje naturalne zainteresowanie Amstradem firm sprzedających oprogramowanie, a użyty popularny Z80 — względną łatwość „przenoszenia" programów np. ze Spectrum na **CPC 6128**.

PODSTAWOWE DANE TECHNICZNE:

Mikroprocesor .... Z 80

Pamięć RAM .... 128 kB

Pamięć ROM .... 48 kB

Organizacja obrazu — trzy tryby pracy
mode 0 — 20×25 znaków
mode 1 — 40×25 znaków
mode 2 — 0×25 znaków

Rozdzielczość ekr. ...... 160×200 pkt.

Klawiatura .... QWERTY

Język .... Basic 1.1 firmy Locomotive Software

Stacja dyskietek 3" w jednej obudowie z komputerem

Podobne względy sprawiły, że bardzo dużo różnych firm zaczęło produkować urządzenia peryferyjne pozwalające na szersze wykorzystanie możliwości posiadanego sprzętu. Można więc dokupić nie tylko drążki sterowe, ale również stereofoniczny syntezator mowy, napędy dysków 3,5 i 5 cali, drukarki z wejściem równoległym Centronics (w tym bardzo dobrą drukarkę firmową DMP 2000), sprzęg szeregowy RS 232, modem, 8-mio wejściowe przetworniki analogowo-cyfrowe, przetwornik cyfrowo-analogowy, sterowniki mocy (np. do sterowania urządzeń zasilanych z 220V), przystawkę TV pozwalającą na odbieranie w kolorze (monitor CTM 664) sześciu programów telewizyjnych, pióra graficzne i tablice graficzne, wzorowaną na Macintosh'u tzw. „myszkę" i wreszcie przystawkę do CPC 6128 symulującą pracę minikomputera firmy IBM — PC.

Pracę na minikomputerze ułatwia również obecność na rynku dużej ilości periodyków poświęconych wyłącznie Amstradowi oraz książek opisujących szczegółowo konstrukcję elektroniczną, oprogramowanie firmowe, systemy operacyjne **AMSDOS** i **CP/M+**, a także propozycje wykorzystania sprzętu wraz z przykładowymi programami.

Przy braku dostępu do literatury na naszym rynku, warto zwrócić uwagę na fakt, że w przypadku posługiwania się magnetofonem nie wszystkie programy z CPC 464 będzie można wykorzystać w 6128. Spowodowane jest to nieco inną organizacją pamięci w tym ostatnim modelu, gdzie obszar używany przez programistów dla 464 przeznaczony jest w **6128** częściowo do obsługi napędu dysków. Warto również podkreślić, że w przypadku wykorzystywania minikomputera do celów profesjonalnych korzystniejsze jest posiadanie zestawu z monitorem monochromatycznym (z zielonym). Praca w trybie 2 tzn. 80 kolumn i 25 wierszy wykorzystywana jest m.in. przez **CP/M** i inne programy użytkowe i w przypadku monitora kolorowego — szczególnie przy pracy z edytorami tekstów — staje się prawie niemożliwa ze względu na nie najlepszą czytelność znaków i szybko postępujące zmęczenie wzroku operatora. Pewnym mankamentem wszystkich modeli Amstrada jest lansowany przez firmę „wewnętrzny" standard dyskietek o średnicy 3 cali. W dodatku droższych niż wprowadzane przez inne firmy dyskietki 3.5 cali.

Wydaje się jednak, że **Amstrad CPC 6128** mimo 8-bitowego mikroprocesora ma jeszcze kilkuletnią przyszłość na świecie minikomputerów 16- i 32-bitowych ze względu na bardzo dobrą grafikę, oprogramowanie i — co dość istotne — względnie niską cenę systemu.

Natomiast wcześniejszy, „przejściowy" CPC 664 kończy powoli swą karierę, przegrywając jedynie o 64 kB pamięci RAM ze swym następcą.

Wojciech Ziółek

MENU -DYSKU-

Jeśli posiadasz wśród swoich dyskietek wiele takich na których nie zamierzasz dokonywać już zmian, to wygodnie będzie Ci umieścić na nich program „MENU DYSKU". Nie będziesz wtedy musiał za każdym razem pisać CAT, żeby zobaczyć jak nazywają się programy i potem wpisywać RUN "nazwa. Wystarczy, że podasz komputerowi rozkaz RUN "DYSK, a on sam wypisze Ci wszystkie nazwy i teraz będziesz mógł wybrać odpowiedni program przez naciśnięcie jednego klawisza. Po wpisaniu listingu musisz w linii 60 umieścić po DATA najpierw liczbę programów, które znajdują się na danej stronie dyskietki, a potem ich nazwy (bez rozszerzenia czyli trzech liter po kropce). Programów tych nie może być więcej niż 9, a programy mogą być wyłącznie typu .BAS lub BIN. W razie potrzeby można zmienić nazwę programu rozkazem I REN, „nowa nazwa", „stara nazwa". Następnie wystarczy włożyć odpowiednią dyskietkę i zapamiętać na niej program rozkazem SAVE "DYSK.

Tomasz Pyć

```
10 REM        *   MENU   D
YSKU  *
20 REM
30 REM    WGRANIE PRZEZ
-RUN"DYSK"
40 '
50 BORDER 1:INK 0,1:IN
K 1,24:PEN 1:PAPER 0:M
ODE 1:LOCATE 11,2:PRIN
T "PROGRAMY NA DYSKU":

60 DATA 4,ARMATY,QUACK
,TASWORD,WALLY
60 DATA 4,ARMATY,QUACK
,TASWORD,WALLY
70 RESTORE 60:READ men
u%:DIM prog$(menu%)
80 FOR x=1 TO menu%:RE
AD prog$(x):LOCATE 2.4
+x*2:PRINT USING"\
   \";prog$(x);:PRINT
STRING$(20,".");x:NEXT
 x
90 in%=VAL(INKEY$):IF
in%<1 OR in%>menu% THE
N 90
100 CLS:RUN prog$(in%)
```

AMX DLA MYSZY

Do nielicznej na razie biblioteki oprogramowania dla „myszy" pracującej z mikrokomputerami Amstrad przybył interesujący i obszerny (na dwóch dyskietkach) pakiet pod nazwą AMX Pagemaker. Jest on kombinacją procesora tekstowego, graficznego i „programatora" drukarki. Umożliwia tworzenie i drukowanie kompozycji tekstowo-graficznych na CPC 464, 664 i 6128. Cena ok. 50 funtów.

EKRAN NA PAPIERZE

W pierwszym numerze francuskiego periodyku **„Micro Strad"** **Joel Jardouin** podaje przykład ciekawego programu umożliwiającego wydruk zawartości ekranu na drukarce. Istnieją wprawdzie programy pisane w Basic'u (w tym również program podany w instrukcji obsługi fabrycznej drukarki Amstrada — DMP 2000), które pozwalają na kopiowanie ekranu, jednak wszystkie mają jedną wspólną wadę — kopiowanie trwa bardzo długo (około 12 minut). Omawiany przez nas program zdecydowanie skraca ten proces do ca 2 minut. Dodatkową jego zaletą jest również możliwość uzyskiwania wydruku znaków definiowanych przez użytkownika, a więc także typowo polskich liter. Jedyną wadą, wspólną zresztą dla wszystkich tego typu programów jest wstrzymywanie wykonywania zasadniczego programu w czas wydruku.

Program ten można wczytać osobno do komputera i po uruchomieniu pisać własny program posługując się w razie potrzeby dokonania wydruku rozkazem CALL& A000 w trybie bezpośrednim. Lepiej jednak, po usunięciu linii od 10 do 50 i linii 330, umieścić go na początku własnego programu i uruchamiać wydruk umieszczając w odpowiednim miejscu programu linię z rozkazem CALL& A000. Należy również pamiętać o tym, że program ten kopiuje CAŁY ekran niezależnie od tego, czy jest zapełniony całkowicie, czy tylko częściowo — a więc wydruk zajmuje tyle samo czasu przy kopiowaniu jednego znaku co i przy przenoszeniu rysunku lub tekstu wypełniającego ekran.

Opr. W.Z.

```
10 '*******************************
20 '* KOPIOWANIE EKRANU GRAFICZ- *
30 '* NEGO NA DRUKARKE TYPU EPSON*
40 '*******************************
50 'Wywolanie programu przez CALL &A000
60 MEMORY &9FFF
70 FOR i=&A000 TO &A0B0
80 READ v:POKE i,v:t=t+v
90 NEXT i
100 IF t<>20961 THEN PRINT "Blad w danyc
h!"
110 DATA 205,166,160,62,27,205,157,160
120 DATA 62,49,205,157,160,205,186,187
130 DATA 205,231,187,50,180,160,17,0
140 DATA 0,33,143,1,34,178,160,62
150 DATA 7,50,177,160,62,10,205,157
160 DATA 160,62,13,205,157,160,62,27
170 DATA 205,157,160,62,76,205,157,160
180 DATA 62,127,205,157,160,62,2,205
190 DATA 157,160,14,0,58,177,160,71
200 DATA 229,197,213,205,240,187,209,193
210 DATA 33,180,160,190,225.55,32,1
220 DATA 167,203,17,43,43,16,233,58
230 DATA 177,160,254,7,40,7.175,203
240 DATA 17,203,17,203,17,121,205,157
250 DATA 160,19,229,33,127,2,55,237
260 DATA 82,225,56,5,42,178,160,24
270 DATA 193,35,124,181,40,32,43,17
280 DATA 0,0,34,178,160,62,7,189
290 DATA 32,146,124,180,32,142,62,4
300 DATA 50,177,160,24,135,205,46,189
310 DATA 56,251,205,43,189,201,62,27
320 DATA 205,157,160,62,64,205,157,160,2
01 ,
330 STOP
```

ZEGAR DLA AMSTRADA

Prawie każdy system komputerowy, duży czy mały, posiada wbudowany zegar. W profesjonalnych systemach zegar ten ustawiany jest przez operatora zaraz po włączeniu maszyny i pracuje przez cały czas, dostarczając, na każde żądanie dowolnego programu, aktualną godzinę i datę. W komputerach domowych zegar maszynowy jest zwykle dużo prostszy. Najczęściej jest to kilka bajtów w pamięci, których wartość zwiększana jest o jeden co pewien określony czas, najczęściej co 0.02 sekundy. Jeśli chcemy uzyskać godzinę czy minutę musimy w każdym programie ustawiać wartości początkowe i potem na podstawie tych bajtów obliczać upływ czasu. Jest to oczywiście bardzo niewygodne i często nawet niemożliwe.

Czytelnicy znający zasady pracy mikroprocesora wiedzą, że w celu zapewnienia w miarę dokładnej pracy zegara wykorzystywany jest mechanizm przerwań, czyli sprzętowego (zewnętrznego) wymuszania skoku do określonego adresu w pamięci komputera. Tam znajduje się procedura, która ma za zadanie wykonanie wszystkich procesów zależnych od czasu (np. czytanie klawiatury, miganie kolorów na ekranie i inne) oraz zwiększenie o jeden wartość naszego zegara.

Ten właśnie mechanizm mogą wykorzystać użytkownicy Amstradów i Schneiderów do dołączenia własnych procedur zależnych od czasu. System operacyjny tych komputerów pozwala na umieszczenie w tak zwanych łańcuchach, informacji o podprogramach maszynowych, które mają być wykonane np. co sekundę. Jeśli teraz mikroprocesor otrzyma sygnał od odpowiedniego zewnętrznego układu scalonego (w Amstradach sygnał ten wysyłany jest przez **Gate Array** co 1/3000 sek.), to nie tylko wykona stałe zadania zawarte w ROMie, ale również przeszuka odpowiednie łańcuchy i sprawdzi, czy nie ma tam jakichś procedur do wykonania.

My wykorzystamy tę zaletę do napisania własnego zegara, który będzie pracował niezależnie od aktualnie wykonywanych programów i będzie podawał czas w normalnej formie (czyli godz.:min.:sek.). Wyposażymy go również w budzik i możliwości sterowania z BASIC-u. Żeby być całkowicie w zgodzie z prawdą należy zauważyć, że nasz zegar, tak jak zresztą i maszynowy, nie pracują, jeśli jakiś program wyłączy przerwania, co jest możliwe za pomocą odpowiedniego rozkazu maszynowego. System i BASIC wyłączają przerwania właściwie tylko w czasie zapisu lub odczytu danych na taśmę lub dyskietkę.

Po wpisaniu i uruchomieniu zamieszczonego obok programu należy odpowiedzieć na pytanie o wersję Amstrada (Schneidera), gdyż od tego zależą adresy pewnych procedur systemowych wykorzystywanych przez program (np. dzielenie przez 10).

Do sterowania programem służą następujące komendy **RSX** (**R**esident **S**ystem **E**xtension) wywoływane poprzez pionową kreskę (**SHIFT** i @) i nazwę:

ustaw,g,m — ustawia wartość początkową zegara g: godzina (1–24), m: minuty (1–59)
alarm,g,m — ustawia czas budzika
wlacz — włącza stałe pokazywanie czasu w prawym, górnym rogu ekranu
wylacz — wyłącza pokazywanie czasu; zegar chodzi
stop — zatrzymuje zegar
start — uruchamia zegar ponownie
**tekst, @ a $** — ustala tekst, z którego pierwsze dziesięć znaków zostanie wypisane na ekranie po uruchomieniu budzika; a$ jest dowolną zmienną tekstową; w 664 i 6128 zamiast można podać od razu tekst w cudzysłowach
**czas, @ a $** — umieszcza z zmiennej tekstowej a$ string z aktualnym czasem w postaci takiej, jak na ekranie; zmiennej a$ należy wcześniej przypisać dowolny tekst o długości 8 znaków (co najmniej).

Po odpowiedzi na pytanie o wersję komputera można naturalnie wyczyścić program rozkazem **NEW** i uruchomić dowolny inny program, który nie zajmuje obszaru pamięci od adresu & A400 do & A5BA. Komendy RSX pozostaną w pamięci i mogą być wykorzystane w innych programach.

Na podstawie „Schneider International" nr 12/1985

oprac. S. Wolicki

```
10 DATA &21,&42,&a4,&11,&0f,&a4,&01,
&00,&8B,&cd,&d7,&bc,&c3,&d7,&a4,&21
20 DATA &3C,&A4,&35,&C0,&36,&32,&23,
&7E,&C6,&01,&27,&77,&FE,&60,&20,&2C
30 DATA &36,&00,&23,&7E,&C6,&01,&27,
&77,&FE,&60,&20,&20,&36,&00,&23,&7E
40 DATA &C6,&01,&27,&77,&FE,&24,&20,
&14,&36,&00,&18,&10,&01,&00,&00,&00
50 DATA &00,&00,&00,&00,&00,&00,&00,
&8B,&0F,&A4,&00,&00,&C5,&D5,&CD,&A6
60 DATA &A4,&3A,&92,&A4,&B7,&28,&0A,
&CD,&65,&A4,&23,&01,&00,&4B,&CD,&93
70 DATA &A4,&D1,&C1,&AF,&C9,&11,&3D,
&A4,&21,&91,&A4,&06,&03,&1A,&4F,&E6
80 DATA &0F,&C6.&30,&2B,&77,&79,&1F,
&1F,&1F,&1F,&E6,&0F,&C6,&30,&2B,&77
90 DATA &13,&2B,&36,&3A,&10,&E7,&C9,
&00,&3A,&30,&30,&3A,&30,&30,&3A,&32
100 DATA &35,&00,&01,&7E,&B7,&C8,&E5
,&C5,&E1,&C5,&CD,&A3,&A4,&C1,&E1,&04
110 DATA &23,&18,&F0,&CF,&47,&93,&11
.&CA,&A4,&21,&3D,&A4,&7E,&FE,&03,&D0
120 DATA &23,&1A,&BE,&C0,&13,&23,&1A
,&BE,&C0,&CD,&C6,&A4,&21,&CC,&A4,&01
130 DATA &01,&46,&CD,&93,&A4,&C9,&CF
,&DD,&94,&C9,&01,&00,&42,&55,&44,&5a
140 DATA &49,&4b,&20,&31,&20,&20,&00
,&21,&E9,&A4,&CD,&D4,&BC,&38,&09,&01
150 DATA &15,&A5,&21,&11,&A5,&CD,&D1
,&BC,&C9,&55,&53,&54,&41,&D7,&41,&4c
160 DATA &41,&52,&cD,&54,&45,&4b,&53
,&D4,&53,&54,&4f,&d0,&57,&4c,&41,&43
170 DATA &da,&57,&59,&4c,&41,&43,&da
,&53,&54,&41,&52,&d4,&43,&5a,&41,&D3
180 DATA &00,&73,&46,&15,&A5.&E9,&A4
,&C3,&43,&A5,&C3,&51,&A5,&C3,&84,&A5
190 DATA &C3,&3C,&A5,&C3,&31,&A5.&C3
,&37,&A5,&C3,&00,&A4,&C3,&9B,&A5,&00
200 DATA &00,&3E,&01,&32,&92.&A4,&C9
,&AF,&32,&92,&A4,&C9,&21,&42,&A4,&CD
210 DATA &DD,&BC,&C9,&FD,&21,&3E,&A4
,&CD,&59,&A5,&FD,&2B,&FD,&36,&00,&00
220 DATA &C9,&FD,&21,&CA,&A4,&CD,&59
,&A5,&C9,&FE,&02,&C0,&26,&00,&DD,&6E
230 DATA &00,&CD,&71,&A5,&FD,&77,&00
,&DD,&6E,&02,&CD,&71,&A5,&FD,&77,&01
240 DATA &C9,&11,&0A,&00,&D7,&b3,&1D
,&7D,&17,&17,&17,&17,&E6,&F0,&6F,&7B
250 DATA &E6,&0F,&B5,&C9,&FE,&01,&C0
,&DD,&66,&01,&DD,&6E,&00,&23,&5E,&23
260 DATA &56,&01,&0A,&00,&21,&CC,&A4
,&EB,&ED,&B0,&C9,&FE,&01,&C0,&DD,&6E
270 DATA &00,&DD,&66,&01,&7E,&FE,&08
,&D8,&E5,&CD,&65,&A4,&E1,&23,&5E,&23
280 DATA &56,&21,&89,&A4,&01,&08,&00
,&ED,&B0,&C9,&00,&00,&00,&00,&00,&00
290 adr=&A400:suma=0
300 koniec=&A5BA
310 IF HIMEM>&A3FF THEN MEMORY &A3FF
320 FOR a=adr TO koniec
330 READ b:POKE a,b:suma=suma+b
340 NEXT a:IF suma<>4836B THEN PRINT
 "Popraw DATA !!!":END
350 INPUT "464,664 czy 6128 (1/2/3):
 ",a%
360 IF a%<1 OR a%>3 THEN 350
370 IF a%=1 THEN POKE &A4A4,&4A:POKE
 &A4C7,&D8:POKE &A574,&CD:POKE &A575
,&C1:POKE &A576,&BD ELSE IF a%=3 THE
N POKE &A4A4,&4B:POKE &A4C7,&E1:POKE
 &A575,&AE
380 CALL &A400
```

LISTA ZMIENNYCH

W wielu przypadkach przy próbach uruchomienia własnego programu okazuje się, że występuje w nim wiele błędów. Stosunkowo łatwe do wykrycia okazują się błędy syntaktyczne (składniowe), gdyż informuje o nich komputer. Jeżeli jeszcze analiza poprawności konstrukcji programu wypada pomyślnie a mimo to program nadal nie działa, to przyczyną może okazać się pomyłka we wpisywaniu nazw zmiennych. Np. jeśli pisząc nazwę zmiennej „suma", która występuje już wcześniej napiszemy „sama", Locomotiv Basic potraktuje ją jako nową, inną zmienną. Przy dużej ilości zmiennych używanych w programie tego typu pomyłka jest dość trudna do wykrycia w stosunkowo krótkim czasie. Załączony program tworzy nową komendę **VARDUMP**, która umożliwia wylistowanie nazw wszystkich zmiennych występujących w tworzonym przez nas programie. Lista taka znakomicie ułatwia wyszukiwanie „przekłamanych" zmiennych. W konstrukcji przedstawionego programu zasługują na uwagę dwa elementy: sposób wpisania danych w liniach od 160 do 340 i występujące w tych liniach po przecinku heksadecymalne „dane pomocnicze". Dane wpisywane są jednym ciągiem, co przyśpiesza wprowadzanie ich z klawiatury, natomiast wybór danych z określeniem ich długości odbywa się w linii 80. „Dane pomocnicze" natomiast reprezentują tak zwaną sumę kontrolną (check sum) czyli wartość, będącą sumą wartości wszystkich danych w linii. W linii 120 realizowane jest sprawdzanie poprawności wpisania danych i w przypadku wystąpienia błędów — generowanie komunikatu o błędzie w danych z podaniem numeru linii, w której błąd miał miejsce. Zabieg taki znacznie przyśpiesza wyszukiwanie niewłaściwie wpisanych danych. W omawianym programie pętla od linii 70 do 110 rozdziela i zapamiętuje dane w grupach dwuznakowych (patrz linia 80), a pętla od linii 50 do linii 130 w każdym kroku sumuje dane z jednej linii i porównuje otrzymaną wartość z sumą kontrolną tej linii, drukując w razie niezgodności (patrz linia 120) komunikat o błędzie.

Przykładowo dla linii 180 odbywa się to przez sumowanie 02+c0+a7+28+ +12+++++++4d+a0 i jeśli wynik jest różny od 3eb drukowany jest komunikat „Blad w danych w linii 180".

Po tych wyjaśnieniach przystępujemy do wpisywania i uruchamiania programu. Przepisujemy program opuszczając linie 140 i 150 zachowując numerację pozostałych linii. Uruchamiamy program. Jeżeli pojawi się napis „blad w danych w linii XXX" — sprawdzamy i poprawiamy tę linię. Jeżeli błędów nie ma, wprowadzamy w trybie bezpośrednim rozkaz **CALL** &a000. Po chwili pojawi się napis OK. Dopisujemy wtedy linie 140 i 150 i zapisujemy program na dyskietkę pod nazwą np. „zmienne". Chcąc skorzystać z programu ładujemy go przez użycie **RUN** „zmienne". Po pojawieniu się napisu OK i Ready wpisujemy własny testowany program i również go uruchamiamy. Po ponownym pojawieniu się napisu Ready lub po przerwaniu programu klawiszem ESCape — np. w przypadku zatrzymania się programu na skutek innych błędów — wywołujemy komendę **! VARDUMP** przez naciśnięcie klawiszy **SHIFT** i @ oraz wpisanie vardump. Po wciśnięciu **RETURN** na ekranie pojawi się nazwa pierwszej zmiennej. Kolejne wciskanie innego dowolnego klawisza będzie powodowało wyświetlanie nazw następnych zmiennych. Pojawienie się w pewnym momencie komunikatu Ready kończy listę zmiennych występujących w analizowanym programie. Wystarczy teraz porównać nazwy występujące w programie, by określić czy nie wprowadziliśmy przypadkowo nazwy zmiennej i jeśli tak się stało — dokonać odpowiedniej korekty.

Wojciech Ziółek

```
10 'Listowanie zmiennych
20 'Komenda wywolujaca zmienne:
30 '"|VARDUMP" i dowolny klawisz
40 MEMORY &9FFF:adress=&A000
50 FOR i=1 TO 19
60 sum=0:READ code$,check$
70 FOR j=1 TO 21 STEP 2
80 byte=VAL("&"+MID$(code$,j,2))
90 POKE adress,byte
100 sum=sum+byte:adress=adress+1
110 NEXT j
120 IF sum<>VAL("&"+check$) THEN PR
INT "Blad w danych w linii ";150+i*
10
130 NEXT i
140 CALL &A000
150 DELETE 10-
160 DATA 214ba0cb4ec0cbce0146a0,565
170 DATA 2142a0cdd1bccd00b9f53a,612
180 DATA 02c0a72812216aae224da0,3eb
190 DATA 2168ae2252a0216cae2292,43a
200 DATA a0f1cd0cb9cdc5a04f4b0d,5fc
210 DATA 0a00c95641524455 4dd000,372
220 DATA 00000003aa0c34ca0002a,2b3
230 DATA 87aee5ed5b85aea7ed52e1,75c
240 DATA ca8fa0eb2323cd9ba03e0d,57d
250 DATA cd5abb3e0acd5abbcd06bb,59a
260 DATA 09e5ed52e138e91818cd9b,5c7
270 DATA a0e5cdc5a028290d0a00e1,500
280 DATA cd06bb235e2356232323l9,30a
290 DATA e5ed5b89aea7ed52e138dd,740
300 DATA c97ef620e67fcd5abbcb7e,6ed
310 DATA 2328f37efe0120083e2501,347
320 DATA 0500c35abbfe0220083e24,367
330 DATA 010600c35abb010800c9e1,392
340 DATA 7ecd5abb23b720f8e90000,53b
```

POWIĘKSZANIE NAPISÓW

Prezentujemy krótki, lecz interesujący program pozwalający na powiększanie liter i znaków graficznych. Program wykorzystuje instrukcję SYMBOL AFTER 32, co pozwala na redefiniowanie wszystkich znaków. Zamiast linii 70, 80 i 85 można wpisać własne i ewentualnie przenumerować linię 90. Zmienne zx% i zy% określają współrzędne lewego górnego rogu, od którego ma się zaczynać napis. Zmienna zc% określa kolor napisu w konkretnej linii. Zmienna zw% określa szerokość znaku, zh% — wysokość znaku a zz$ — tekst, który ma być drukowany. Ten podprogram można również włączyć do własnego programu po usunięciu wszystkich komentarzy i linii od 70 do 90 oraz nadaniu linii 60 numeru najniższego w programie np. 1. Następnie w odpowiednim miejscu własnego programu należy wpisać własny tekst wzorując się na konstrukcji linii od 70 do 85 pamiętając o zakończeniu jej komendą GOSUB 40000.

(wz)

```
60 CALL &BC02:BORDER 6:MODE 0:SYMBOL AFT
ER 32:DEF FNz1=zy%-z3%*zh%:DEF FNz2=zy%-
z3%*zh%+zh%:DEF FNz3=zx%+(7-z5%)*zw%+z1%
*zw%*8:DEF FNz4=zx%+(8-z5%)*zw%+z1%*zw%*
8-8:DEF FNz5=z4% AND 2^z5%:CLS
70 zz$="Przyklad":zx%=2:zy%=398:zc%=1:zw
%=10:zh%=12:GOSUB 40000
80 zz$=" dzialania ":zx%=6:zy%=250:zc%=7
:zw%=7:zh%=6:GOSUB 40000
85 zz$="programu":zx%=40:zy%=170:zc%=11:
zw%=9:zh%=22:GOSUB 40000
90 GOTO 90
40000 FOR z1%=0 TO LEN(zz$)-1:z2%=ASC(MI
D$(zz$,z1%+1,1))-32:FOR z3%=1 TO 8:z4%=P
EEK(HIMEM+z3%+z2%*8):FOR z5%=7 TO 0 STEP
 -1:IF FNz5 THEN ORIGIN 0,0,FNz3,FNz4.FN
z1,FNz2:CLG zc%
40010 NEXT z5%,z3%,z1%:RETURN
```

Klan Amstrad — Schneider redagują:
Tomasz Pyc, Sergiusz Wolicki

INTERFACE TYPU „KEMPSTON" DO MANIPULATORA

DLA ZX SPECTRUM

Użytkownicy komputera SPECTRUM zetknęli się zapewne niejednokrotnie z grami (lub innymi programami), w których na wstępie należy wybrać sposób sterowania ruchem obiektu na ekranie. Sterowanie za pomocą klawiszy jest niewygodne, a oprócz tego niszczy klawiaturę. Wielkim udogodnieniem jest tu „joystick", czyli mówiąc po polsku drążek sterowy. Efekty zastosowania manipulatora są widoczne natychmiast (na przykład w postaci większej ilości punktów zdobytych w grze).

Jednym ze standardowych interface'ów do manipulatora jest produkt typu brytyjskiej firmy KEMPSTON, opcja wyboru interfejsu tego typu jest umieszczana w większości firmowych programów. Cena urządzenia w Wielkiej Brytanii wynosi około 15 funtów. Proponujemy czytelnikom „BAJTKA" posiadającym żyłkę majsterkowicza samodzielne wykonanie interface'u w pełni odpowiadającego standardowi KEMPSTON, o koszcie nie przekraczającym 1500 zł. Koszt ten uwzględnia konieczność zakupu potrzebnych części na wolnym rynku, gdyż nie są one produkowane w kraju.

Schemat ideowy urządzenia, rysunek obwodu drukowanego, sposób montażu elementów, a także sposób podłączenia manipulatora są pokazane na rysunkach i nie wymagają dodatkowych wyjaśnień. Należy tylko dodać, że przy obsłudze interface'u jest wykorzystywana instrukcja IN 223 mikroprocesora Z80, przy której szyna adresowa A5 = 0. Poszczególne bity danych mają wówczas następujące znaczenie:

D0 — w prawo
D1 — w lewo
D2 — w dół
D3 — w górę
D4 — strzał

Informacje te można wykorzystać przy pisaniu własnych programów, w których chcemy zastosować manipulator. Instrukcja IN istnieje również w języku BASIC, a więc programy takie mogą być pisane także przez mniej zaawansowanych użytkowników SPECTRUM.

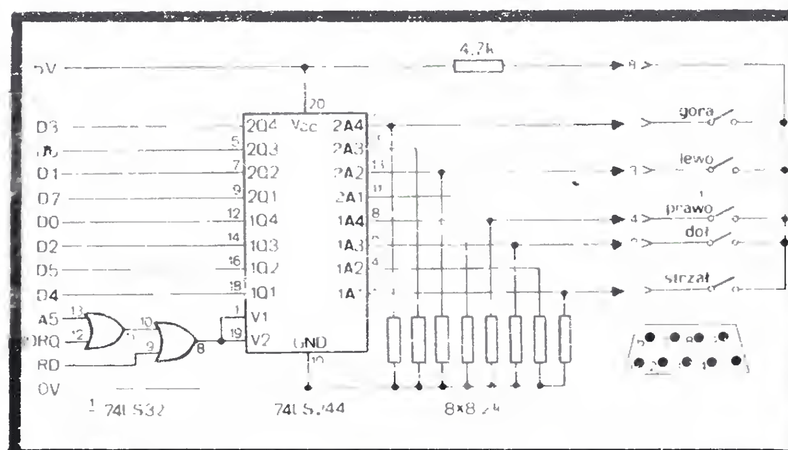
Sposób wykonania manipulatora pozostawiamy inwencji czytelników, potrzebne będą do tego wyłączniki z zestykami chwilowymi, odpowiednia obudowa i trochę cierpliwości przy montażu. Kabel wychodzący z manipulatora powinien być zakończony gniazdem, wtyk jest zamontowany w interface'u.

Można też po prostu kupić gotowy manipulator na jednej z wielu giełd sprzętu komputerowego, lecz wydatek jest dość duży.

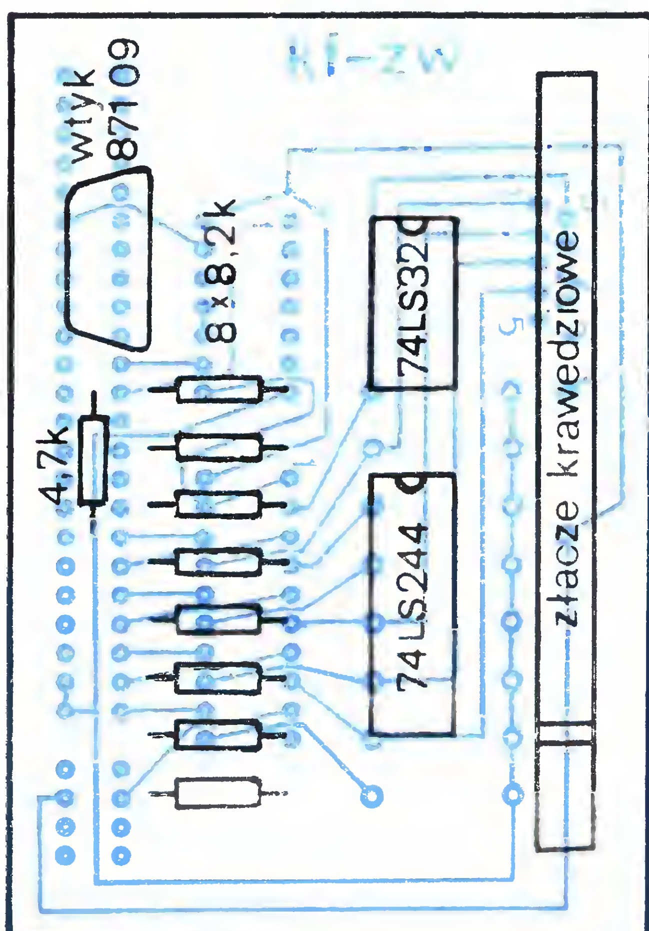
Na zakończenie jedna uwaga, o której zawsze należy pamiętać przy podłączaniu czegokolwiek do SPECTRUM: wtyczka zasilająca powinna być wyjęta!

Wykaz części:
układy scalone: 74LS244 (1 szt.), 74LS32 (1 szt.),
rezystory: 8,2 k (8 szt.), 4,7 k (1 szt.),
złącze CANON typ 871/881 9 styków (1 komplet),
złącze krawędziowe 2×28 styków (1 szt.).

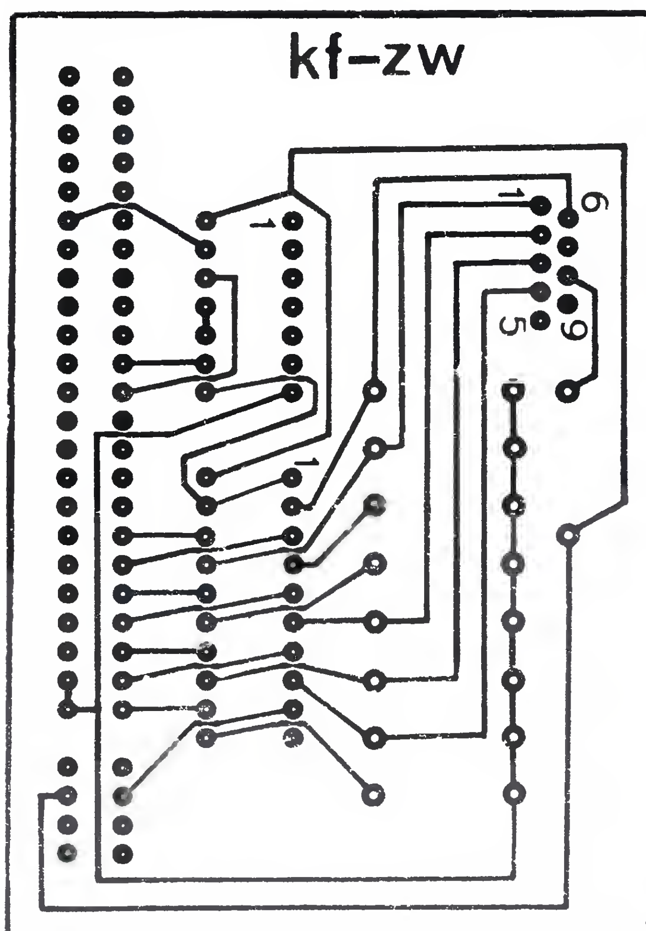
Konrad Fedyna
Zygmunt Wereszczyński



Rys.1. Schemat ideowy interface'u i sposób podłączenia manipulatora.



Rys.2. Rysunek płytki drukowanej.



Rys.3. Sposób rozmieszczenia elementów na płytce.

JAK SCHOWAĆ OBRAZEK

Przekonajmy się, że język maszynowy nie jest tak straszny jak starają się sugerować niektórzy „wtajemniczeni". Spróbujemy dzisiaj wspólnie napisać prosty program zapamiętywania obrazu. Będziemy też mogli dzięki niemu zobaczyć różnicę w szybkości działania programu w BASIC-u i w kodzie maszynowym.

Pamięć obrazu to 6144 bajty informacji o tym, które punkty są w kolorze papieru, a które atramentu, oraz 768 bajtów tzw. atrybutów, czyli informacji o kolorach papieru i atramentu, a także jasności **(BRIGHT)** i migania **(FLASH)**.

Zawartość pamięci może być modyfikowana przy pomocy **POKE** i odczytywana przez **PEEK**. Pamięć obrazu nie różni się niczym od innych obszarów pamięci operacyjnej systemu. Oczywiście operacji na ekranie wygodniej jest dokonywać przy użyciu instrukcji **PRINT** i **SCREEN$**.

Zawartość pamięci ekranu można przepisać do innego obszaru pamięci, gdzie nie zostanie ona zniszczona przez RUN, CLEAR czy CLS. Umożliwia to poniższy program:

```
10 FOR A = 16384 TO 23295
20 POKE A + 9466, PEEK A
30 NEXT A
```

Najpierw trzeba jednak wyznaczyć obszar w pamięci komputera, gdzie umieszczona zostanie kopia pliku obrazu. W tym celu piszemy **CLEAR 25849.** Uwaga: instrukcja ta musi poprzedzić zapełnianie obrazu! W pierwszym wypadku **CLEAR** skasuje całą zawartość ekranu. Z tego samego powodu uruchamiać ten program należy nie instrukcją **RUN**, lecz **GOTO 10**.

Kiedy komputer zakończy przepisywanie ekranu do pamięci operacyjnej obrazek jest bezpieczny. Teraz nie „zaszkodzi" mu nawet **NEW**, które kasuje pamięć tylko do miejsca wskazanego przez **CLEAR**.

Powtórne umieszczenie obrazu na ekranie umożliwia następny program:

```
10 CLS
20 FOR A = 16384 TO 23295
30 POKE A, PEEK A + 9466
40 NEXT A
```

Nietrudno zauważyłeś, że szybkość działania tego programu nie jest „rewelacyjna". Niestety BASIC — mimo całej swej prostoty i przejrzystości działania — nie należy do najszybszych. Błyskawiczne przepisywanie obrazu umożliwia program w języku maszynowym.

Język maszynowy jest to zbiór rozkazów do bezpośredniego wykonania przez jednostkę centralną komputera czyli mikroprocesor. Tymczasem BASIC nie jest wykonywany bezpośrednio przez mikroprocesor, lecz jest przekładany na język maszynowy przy pomocy programu w ROM'ie. Może jednak podać bezpośrednio do mikroprocesora tzw. kody maszynowe, czyli rozkazy do bezpośredniego wykonania.

Aby zrozumieć w jaki sposób mikroprocesor wykonuje programy maszynowe musimy zapoznać się z pojęciem „rejestry mikroprocesora".

Rejestry są czymś w rodzaju 8-bitowej mikropamięci. Przy ich pomocy procesor komunikuje się z otoczeniem tj. pamięcią i układami **I/0** (wejścia/wyjścia). W mikroprocesorze **Z80** programowo są dostępne rejestry: **A** — czyli **akumulator**, B, C, D, E, H, L oraz rejestr wskaźników **(flag) F**. Każdemu z tych rejestrów odpowiadają rejestry pomocnicze, oznaczone jako **A, B, C, D, E, H, L, F.** Pojedyncze re-

jestry można łączyć również w 16-bitową parę rejestrów. Takimi parami są **AF**, **BC**, **DE**, **HL** i odpowiednio **AF**, **BC**, **DE**, **HL**. Poza tym są jeszcze 16-bitowe rejestry specjalne. 2 rejestry indeksowe **IX** i **IT**, rejestr wskaźnika stosu **SP**, rejestr licznika rozkazów **PC** oraz dwa 8-bitowe rejestry — wektora przerwań **I** oraz odświeżania pamięci **R**. Oczywiście **Z80** zawiera jeszcze inne rejestry pmocnicze, które jednak nie są dostępne bezpośrednio.

W języku maszynowym każdemu rozkazowi odpowiada określony kod liczbowy — stąd też nazwa kod maszynowy. Aby można było łatwo operować takimi rozkazami każdemu z nich jest przyporządkowany tzw. mnemonik, czyli symbol literowy, który ułatwia jego zapamiętanie. I tak np. rozkazowi załadowania (wszystko jedno czy pamięci czy rejestru) odpowiada mnemonik **LD**, rozkazowi powrotu **RET** itp. Do procesora trzeba jednak podać kody maszynowe i do tego celu służy program assembler, który tłumaczy mnemoniki na kody maszynowe. Jeżeli nie posiadasz takiego programu możesz zrobić to inaczej, ale pod warunkiem, że program maszynowy nie jest zbyt długi. Odpowiadające mnemonikom kody możesz odszukać np. w dodatku A instrukcji do ZX Spectrum na str. 135 i następnych. I tak, na przykład rozkazowi **LDHL, nn** odpowiada kod 33, gdzie nn jest dwubajtową liczbą z zakresu 0 — 65535, rozkazowi **REN** — 201 itd.

W języku wewnętrznym mikroprocesora Z80 analogiczny program jak podany na początku będzie wyglądał następująco:

LD HL, 16384; adres początku obrazu
LD DE, 25850; adres początku kopii
LD BC, 6912; liczba bajtów przesłania
LDIR; przesłanie blokowe
RET; powrót do BASIC-u

W programie tym rozkazy **LD** oznaczają załadowanie par rejestrów mikroprocesora podanymi liczbami. Rozkaz **LDIR** jest to tzw. przesłanie blokowe, które polega na tym, że zawartość komórki pamięci o adresie określonym zawartością pary **HL** zostaje przepisana do komórki o adresie określonym parą **DE**. Po każdym przesłaniu bajtu zawartości par HL i DE są powiększane przez procesor o 1. Para BC stanowi licznik wskazujący, ile jeszcze bajtów pozostało do przesłania; po każdym przesłaniu bajtu jest zmniejszana o 1. Procesor powtarza przesłania aż do wyzerowania pary BC. Rozkaz powrotu **RET** powoduje powrót do **BASIC-u**.

W ostatniej kolumnie podajemy kody rozkazów w zapisie dziesiętnym. Liczby dwubajtowe są zapisywane w dwóch komórkach pamięci. W drugiej umieszczamy wynik dzielenia przez 256, a w pierwszej resztę z tego dzielenia.

Program ten możesz wpisać do pamięci przy pomocy programu w BASIC-u:

```
10 CLEAR 25499
20 FOR A = 25500 TO 25511
30 READ B: POKE A, B
40 NEXT A
50 DATA 33,0,64,17,250,100,1,0,
27,237,176,201
```

Uruchom ten program, a potem umieść jakiś obraz na ekranie. Teraz wywołaj program maszynowy instrukcją **RANDOMOMIZE USR 25500**. Po powrocie z programu maszynowego obraz zapisany jest w pamięci. Aby znów wywołać go na ekran przy użyciu programu maszynowego, należy zamienić zawartości par **HL i DE**

Wyglądać to będzie tak:

LD HL,25850; adres początku kopii
LD DE,16384; adres początku obrazu
LD BC,6912; liczba bajtów przesłania
LDIR; przesłanie blokowe
RET; powrót do BASIC-u

W programie w BASIC-u wystarczy zamienić linię 50:

```
50 DATA 33,250,100,17,0,64,1,0,
27,237,176,201
```

Następnie piszesz **RANDOMIZE USR 25500** i natychmiast możesz ponownie zobaczyć poprzedni obraz na ekranie.

W przypadku wykonywania przesłania w BASIC-u można obserwować efekt jego działania, obraz jest wykreślany linia po linii. W przypadku programu maszynowego nie daje się zauważyć etapów pośrednich kreślenia obrazu — pojawia się on natychmiast, o czym czytelnicy mogą się przekonać.

Artur Szymański (lat 17)



Dzisiaj w Klanie Spectrum prezentujemy dwa programy realizujące w kodzie maszynowym podobny problem. Wszystkim stawiającym pierwsze kroki w programowaniu w języku wewnętrznym proponujemy dokładne przeanalizowanie obu przykładów.

(red)

ZAPIS OBRAZU ZA RAMTOP

W niektórych grach stosuje się bardzo efektowny sposób rysowania grafiki na ekranie podczas ładowania. Nie powstaje ona linia po linii, lecz ukazuje się na ekranie momentalnie, już pokolorowany. Więksi znawcy komputera mówią wtedy, że „SCREEN wyszedł zza RAMTOPU". Jest to prawda, lecz należy wytłumaczyć, jak to się robi. RAMTOP (szczyt RAM-u) to ostatni adres dostępny dla programu pisanego w BASIC-u. Program ten zaczyna się od adresu 23755 (wyższy bajt numeru pierwszej linii) a kończy się na adresie 23755 + długość programu. Jeżeli adres końcowy przekroczy RAMTOP to pojawia się komunikat „Out of memory" (brak pamięci). Wartość RAMTOP-u ustawia się instrukcją CLEAR nowy RAMTOP, np. CLEAR 30000. Jeśli zadana wartość RAMTOP-u będzie za mała (taka, że nie mieści się program w BASIC-u pomiędzy 23755 a RAMTOP-em) to pojawia się komunikat „RAMTOP no good" Komenda NEW kasuje pamięć do RAMTOP-u. Teraz musimy ustawić RAMTOP na 29999 (linia 10), żeby od 30000 pisać program w kodzie maszynowym (linie 30—50) i wywołać obraz przez RANDOMIZE USR 30000 Dane z pamięci obrazu umieszczamy od adresu 30012 (linia 20) Można to też wykonać przez: LOAD "" CODE 30012.

Znaczenie poszczególnych bajtów kodu maszynowego:

— **17,0,64, ld de,16384** do pary rejestrów de ładuje adres pierwszego bajtu pamięci ekranu

— **33,60,117 ld hl,30012** do pary rejestrów hl ładuje adres pierwszego bajtu danych przeniesionego z pamięci ekranu

— **1,0,27 ld bc,6912** do pary rejestrów bc ładuje długość pamięci ekranu

— **237 after ED** przejście w tryb ED odczytywania rozkazów

— **176 ldir** wymiana bajtów pomiędzy adresem zawartym w parze rejestrów hl a adresem w parze rejestrów de z uwzględnieniem ich ilości — liczba w parze rejestrów bc

— **201** ret powrót do BASIC-u

Przykład użycia programu:

— w liniach 1000—9998 napisać program realizujący na ekranie grafikę, a w linii 9999 wpisać GO TO 20.

— linię 10 zmienić na: 10 CLEAR 29999 : GO TO 1000

— zamiast linii 20 można wpisać: 20 LOAD "" CODE 30012 i ładować z taśmy SCREEN$ (nic nie będzie się rysowało, ale efekt jest taki sam)

UWAGA: procedura w kodzie maszynowym nie jest relokowalna.

Marcin Przasnyski (lat 16)

```
10 CLEAR 29999
20 FOR f=16384 TO 23296: POK
E 13628+f, PEEK F: NEXT f
30 FOR n=30000 TO 30011
40 READ d: POKE n,d
50 NEXT n
60 PRINT "Wcisnij dowolny kl
awisz": PAUSE 0
70 RANDOMIZE USR 30000
100 DATA 17,0,64,33,60,117,1
,0,27,237.176,201
```

KODY KOMPUTEROWE — KODY ZNAKOWE

Pamięć komputera jest zorganizowana w komórki (bajty) po osiem bitów każda. Postać informacji w bajtach jest zawsze taka sama, tzn. ciąg zer i jedynek logicznych. Powstaje zatem duży problem interpretacji tych danych i rozróżnienia, które co sobą przedstawiają. Jeden bajt reprezentuje sobą zawsze jakąś liczbę w przedziale od zera do dwustu pięćdziesięciu pięciu zapisaną w kodzie wewnętrznym maszyny, w systemie dwójkowym. Liczbę tę możemy odczytywać na wiele sposobów. Każdy z nich ma jakieś zalety. Przeważnie różnice polegają na zmianie bazy systemu liczbowego, skąd otrzymujemy odczyt w systemie szesnastkowym (heksadecymalnym), dziesiątkowym (decymalnym), ósemkowym (oktalnym), dwójkowym (binarnym).

Można się umówić, że liczba w jednym bajcie z zakresu od 0-255 nie przedstawia samej siebie, a reprezentuje kod jakiegoś znaku, np. jednej z liter alfabetu. Gdy umowa ta będzie na tyle powszechna, że przestrzegać jej będzie większość producentów hardware'u (np. monitorów, drukarek i samych komputerów) i software'u, to powyższa umowa stanie się pewnym standardem.

Kodem takim jest popularny, używany obecnie na całym świecie kod ASCII (American Standard Code for Information Interchange) lub pewna jego odmiana znana u nas pod nazwą ISO-7. Jest to kod oparty o siedem bitów z ośmiu w bajcie, zatem przypisujący różne kody aż 128 różnym znakom (na siedmiu bitach zakodować można właśnie tyle różnych stanów — 2^7=128). Drugie 128 zostawione jest do indywidualnej dyspozycji na znaki nie należące do standardu, np. jakieś specjalne znaki graficzne lub dodatkowe litery alfabetu, charakterystyczne dla danego języka (np. ą, ę, ż, ć itd.). Znaki należące do kodu ASCII posiadają numery od 0-127, a numery od 128-255 są właśnie tymi pustymi.

Kod ASCII jest podzielony na dwie części. Jedna (od 0 do 32) zawiera znaki, które nie mają reprezentacji graficznej, druga (od 33 do 127) — znaki z postacią graficzną. Znaki bez reprezentacji graficznej są to tak zwane znaki sterujące. Mają one pewne umowne znaczenie i kontrolują wyprowadzanie lub wprowadzanie danych do komputera.

Tabela zawiera znaki kodu ASCII i postać binarną każdego znaku.

Znaki sterujące to:

NUL: (null) znak pusty (zero)
SOH: (Start of heading) początek nagłówka
STX: (Start of text) początek tekstu
ETX: (End of text) koniec tekstu
EOT: (End of transmission) koniec transmisji
ENQ: (Enquiry) zapytanie
ACK: (Acknowledge) potwierdzenie
BEL: (Bell) dzwonek lub sygnał dźwiękowy
BS: (Backspace) krok do tyłu
HT: (Horizontal Tabulation) tabulacja pozioma
LF: (Line Feed) wprowadzanie linii
VT: (Vertical Tabulation) tabulacja pionowa
FF: (Form Feed) wprowadzenie strony
CR: (Carriage Return) powrót karetki
SO: (Shift Out) SHIFT „zwolniony"
SI: (Shift In) SHIFT „wciśnięty"
DLE: (Data Link Escape) „ucieczka" na łącza danych
DC1: (Device Control 1) sterowanie urządzenia 1
DC2: (Device Control 2) sterowanie urządzenia 2
DC3: (Dewice Control 3) sterowanie urządzenia 3
DC4: (Device Control 4) sterowanie urządzenia 4
NAK: (Negative Acknowledge) potwierdzenie negatywne
SYN: (Synchronous Idle) synchronizm „na luzie"
ETB: (End of Transmission Block) koniec bloku transmisji
CAN: (Cancel) usuń
EM: (End of Medium) koniec nośnika
SUB: (Substitute) wymień
ESC: (Escape) ucieczka
FS: (File Separator) separator zbiorów
GS: (Group Separator) separator grup
RS: (Record Separator) separator wierszy
US: (Unit Separator) separator jednostkowy
≡: (Delete) wymaż
SP jest znakiem graficznym oznaczającym puste miejsce (spacja)

Znaki te realizują funkcje, których działanie oddają ich nazwy. Oto przykładowe zastosowanie dwóch z nich.

Większość zbiorów tekstowych, jak np. tekst programu (w dowolnym języku), jest napisanych wiersz po wierszu. Oznacza to, że koniec każdego z nich musi być jakoś oznaczony, aby urządzenie wyświetlające (drukujące) wiedziało kiedy zmienić linię. Na końcu każdej linii występują zatem dwa znaki sterujące: CR realizuje powrót kursora na ekranie (karetki drukarki) do początku danej linii, a LF zmienia linię tzn. ustawia kursor (karetkę) w nowej linii.

Tabelkę odczytuje się w następujący sposób:

b7..b1 kolejne bity w kolejności od najstarszego do najmłodszego, określające kod znaku alfanumerycznego. Znak lub — jego kod znajdujemy odczytując odpowiedni wiersz i kolumnę tabelki. Np. znak VT ma kod 0001011 i oczywiście bit ósmy jest również 0, a kod 1101100 odpowiada literze „l".

W kodzie ASCII zapisywanych jest większość zbiorów z którymi mamy do czynienia, jak np. zbiory tekstowe, programy, zbiory danych. Znaki wyświetlane na ekranie TV lub monitora, wysyłane na drukarkę, poprzez odpowiedni interface, i przesyłane między komputerami kodowane są przeważnie właśnie w ASCII.

		b7	0	0	0	0	1	1	1	1
		b6	0	0	1	1	0	0	1	1
b b b b		b5	0	1	0	1	0	1	0	1
4 3 2 1	rząd	kol	0	1	2	3	4	5	6	7
0 0 0 0	0		NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1		SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2		STX	DC2	„	2	B	R	b	r
0 0 1 1	3		ETX	DC3	#	3	C	S	c	s
0 1 0 0	4		EOT	DC4	$	4	D	T	d	t
0 1 0 1	5		ENQ	NAK	X	5	E	U	e	u
0 1 1 0	6		ACK	SYN	&	6	F	V	f	v
0 1 1 1	7		BEL	ETB	'	7	G	W	g	w
1 0 0 0	8		BS	CAN	(	8	H	X	h	x
1 0 0 1	9		HT	EM	)	9	I	Y	i	y
1 0 1 0	10		LF	SUB	*	:	J	Z	j	z
1 0 1 1	11		VT	ESC	+	;	K	[	k	{
1 1 0 0	12		FF	FS	,	<	L	\	l	|
1 1 0 1	13		CR	GS	–	=	M	]	m	}
1 1 1 0	14		SO	RS	.	>	N	∧	n	~
1 1 1 1	15		SI	US	/	?	O	–	o	≡

Drugi kod, na który chciałbym zwrócić uwagę mimo, iż jest rzadziej używany, to RADIX-50. W odróżnieniu od kodu ASCII, gdzie jeden znak zajmuje jeden bajt, w RADIX-50 w dwóch bajtach możemy zapisać trzy znaki. Odbywa się to kosztem ich ilości. W porównaniu z ASCII jest ich znacznie mniej bo tylko 39 plus jeden nie używany. Jest to nic innego, tylko system liczbowy o podstawie 40 (oktalnie jest to właśnie 50 stąd nazwa RADIX-50). 16-bitową wartość kodującą 3 znaki otrzymujemy ze wzoru:

(i ✱ 50 + j) ✱ 50 + k

Jest to wzór zapisany w systemie oktalnym, gdzie i,j,k przedstawiają wartości trzech znaków w kodzie RADIX. Jak widać kod danego znaku ma różną wartość w zależności od miejsca na którym się znajduje. Znaki występujące w tym kodzie to SP, A-Z, **$**,., 0-9 i w tej kolejności przyjmują wartości od 0-47 oktalnie (kod 35 jest nie używany). Np. ciąg trzech liter X2B przyjmie wartość:

```
X =  30✱ 50 ✱50 = 113000
2 =      40 ✱50 =   2400
B =           2 =      2
                  ------
                  115402
```

Maksymalna liczba w kodzie RADIX-50 t.j. 174777 odpowiada ciągowi 999 (kod „9" to 47, czyli 47✱50✱50+47✱50+47). Nie należy zapominać, że wszystkie obliczenia wykonujemy w systemie oktalnym).

Kod RADIX-50 jest dość powszechnie używany na komputerach PDP i SM.

Na zakończenie chciałbym zwrócić uwagę na jeszcze jeden kod, z którym posiadacze domowych komputerów mają styczność na codzień. Polecam rozgryzienie tego kodu w ramach własnych wprawek bez zaglądania do „manual'a" komputera. Jest to kod wszystkich słów kluczowych w BASIC'u. W celu ograniczenia zajętości pamięci przez program napisany w BASIC'u wszystkim słowom kluczowym przyporządkowane są jednobajtowe kody, i w ten sposób jest ten program pamiętany. Do wykonania tego zadania wystarczy tylko w zmyślny sposób wykorzystać instrukcję PEEK. Ponadto sami na własny użytek możemy wymyślać zupełnie inne oryginalne kody alfanumeryczne i w ten sposób kodować własne zbiory, zabezpieczając je przed niepożądanym odczytaniem.

Pomyślnej zabawy w ENIGMĘ!

Krzysztof Czernek

POKE rzysta

Dzisiaj chcę Wam zaprezentować drugą część POKE'ów do gier dla komputera ZX-Spectrum. Dla tych, którzy z terminem tym nie zetknęli się jeszcze, wyjaśniam, że POKE x,y jest to rozkaz, który do komórki pamięci x ładuje wartość y. Amatorzy gier komputerowych często korzystają z tej instrukcji aby ułatwić sobie zadanie. (np. unieśmiertelnić naszego ludzika).

Można to zrobić tylko wtedy, gdy program nie jest specjalnie zablokowany. Jeżeli np. podano, że na Knight Lore działa POKE 53567,0, to należy załadować program Knight Lore (ładujący), wcisnąć BREAK, odszukać ostatni RANDOMIZE USR w programie i tuż przed nim wpisać w linię podany POKE. Teraz wystarczy RUN lub CONTINUE.

Trudniej jest, gdy po wciśnięciu BREAK program się blokuje. Wtedy trzeba próbować ładować pilota przez MERGE" ". Istnieje jednak duża liczba gier, które zabezpieczone są w taki sposób (np. programem w języku wewnętrznym) iż wprowadzenie tego rozkazu do komputera jest niemożliwe. W tym przypadku najprostszą i w zasadzie jedyną możliwością jest użycie któregoś z programów kopiujących — ja używam do tego celu polskiego programu „COPY–COPY". Po naładowaniu programu kopiującego, należy za pomocą komendy LOAD załadować całą grę i przy pomocy rozkazu POKE wpisać w odpowiednie komórki podane wartości. Teraz wystarczy za pomocą SAVE nagrać całość na kasetę i już posiadamy wersję gry z nieskończoną ilością ludzików.

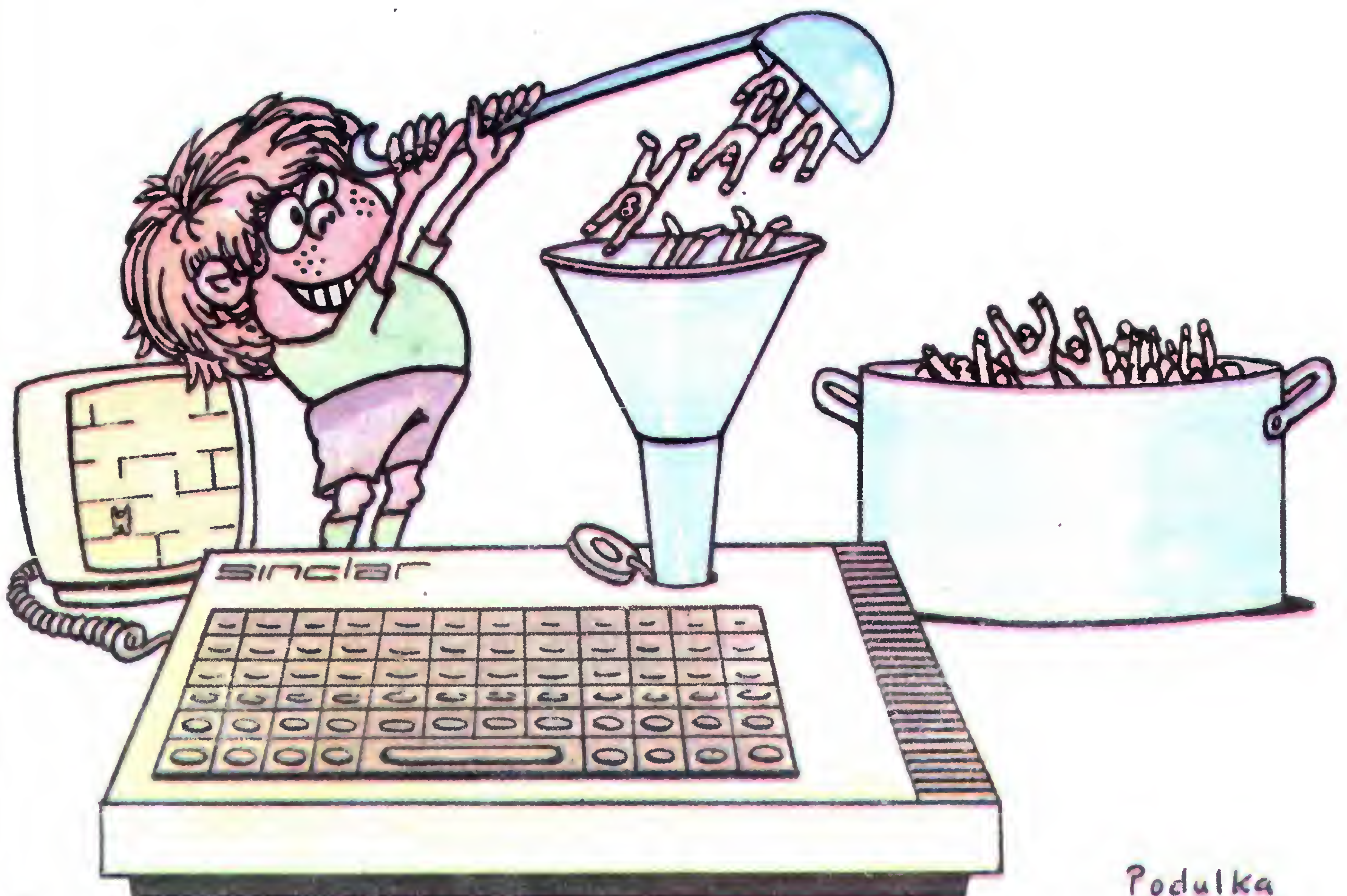
A teraz już POKE'i:

Knight Lore — POKE 53567,0
Defender — POKE 30822,255 : POKE 37815,255
Sabre Wulf — POKE 43575,255
Ground Attack — POKE27872,0
Mutant Monty — POKE 54933,0
Cavern Fighter — POKE 31683,0 : POKE 31684,0
Black Hawk — POKE 34695,183
River Rescue — POKE 33199,255
Atic Atac — POKE 36519,0
Jet Set Willy — POKE 39899,0
Moon Alert — POKE 39754,0
Scuba Dive — POKE 55711,x (ludziki)
Lunar Jetman — POKE 36966,224 POKE 36945,3
The Pyramid — POKE 44685,0
Zooom — POKE 24743,0
Hunchback — POKE 26888,0
Mr. Wimpy — POKE 33693,0
Deathchase — POKE 26463,0
Zaxxon — POKE 48825,x (ludziki)
Wizard's Lair — POKE 25522,x (ludziki)
Underwurlde — POKE 59376,0 POKE 59380,0
Kokotoni Wulf — POKE 43742,0
TLL — POKE 55006,0
Bruce Lee — POKE 51795,0
Booty — POKE 58294,0
Airwolf — POKE 45982,0
Alchemist — POKE 47340,0
Project Future — POKE 27662,0
Alien 8 — POKE 51736,0
Gyroscope — POKE 54754,200

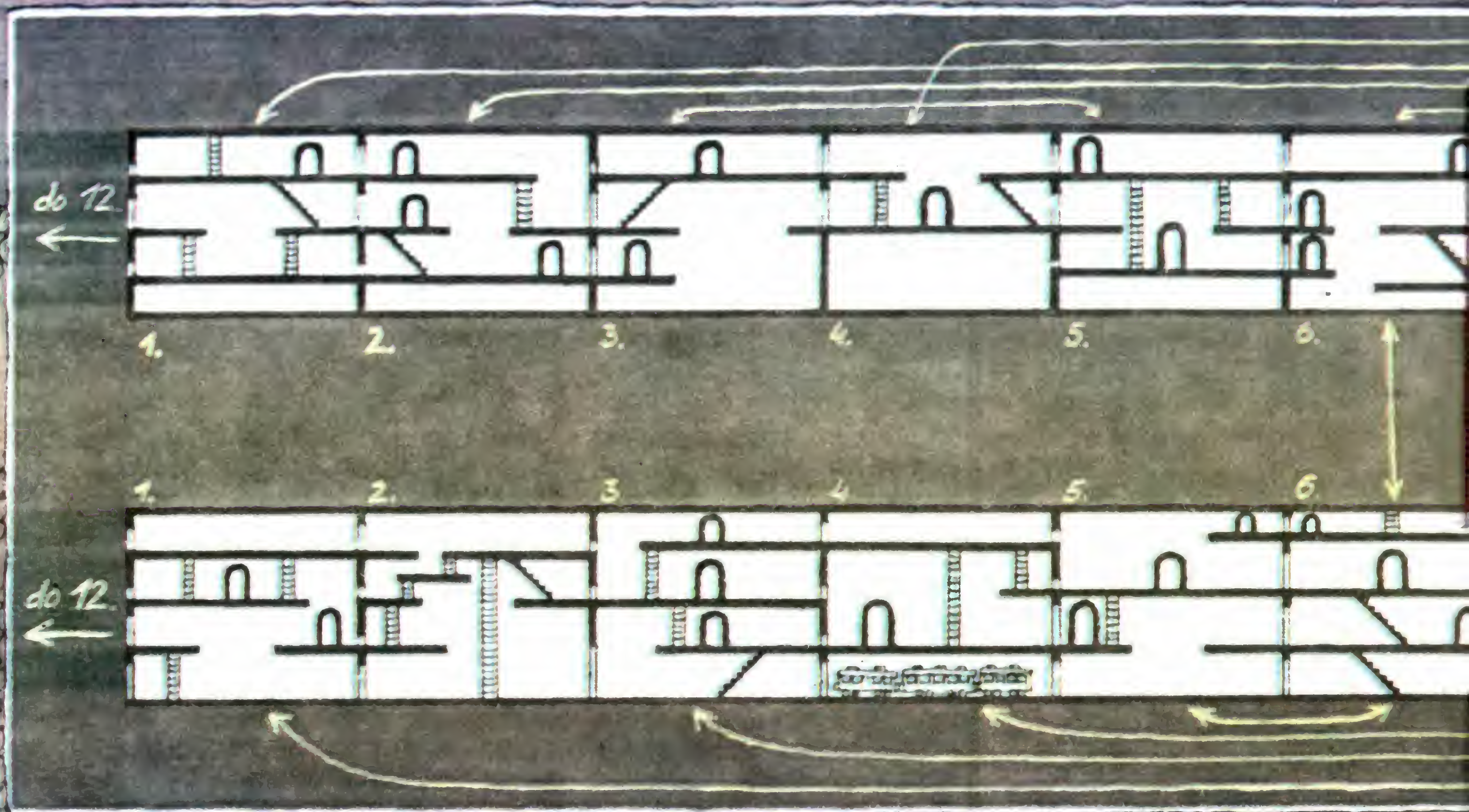
Przy POKE'ach, gdzie znajduje się x, należy zamiast niego podać wartość między 1 a 255. A teraz grajcie...

WESOŁEJ ZABAWY!!!

GRACZ



ROLAND'S RAT RACE



WYŚCIG SZCZURA ROLANDA

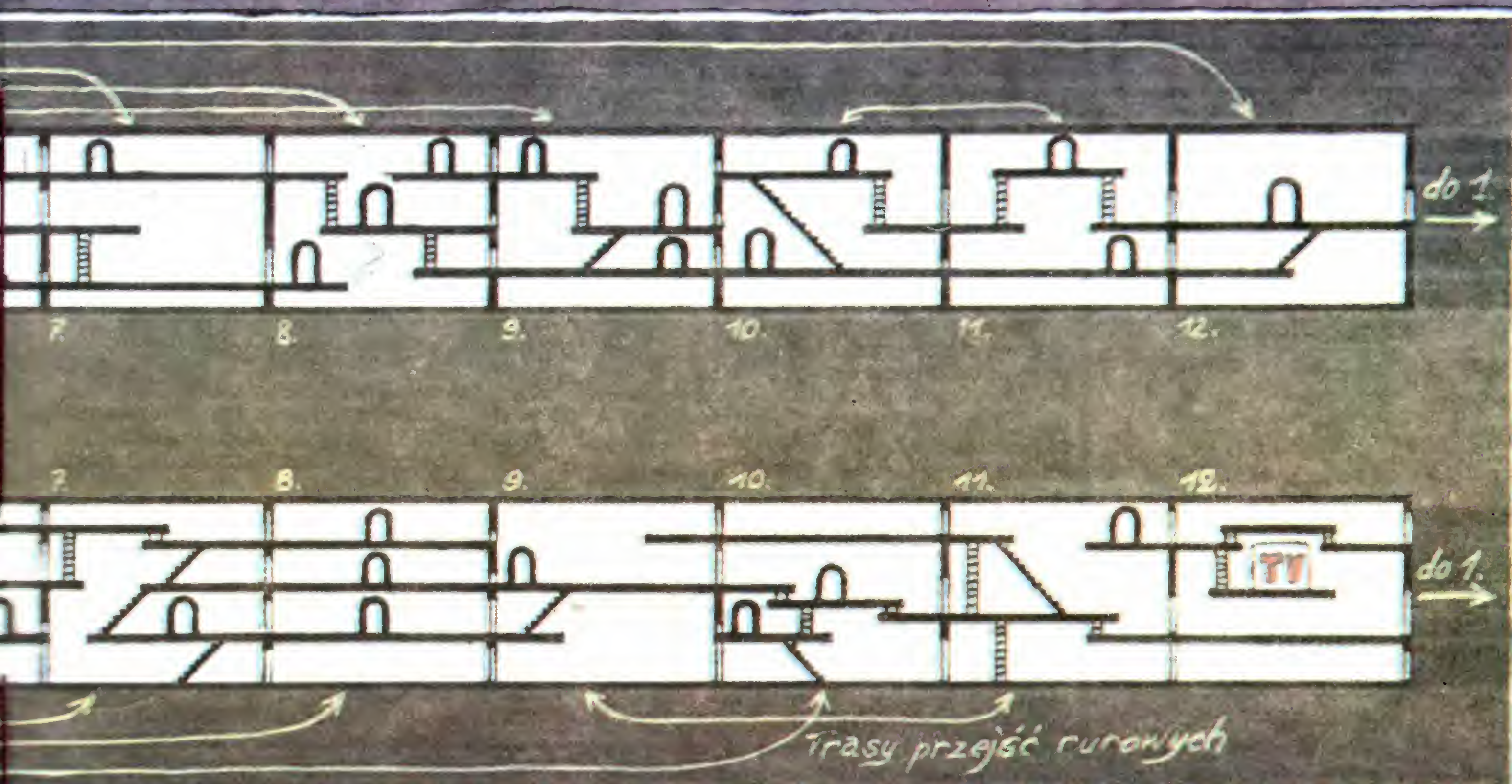
Jeżeli chcesz dostać się do studia telewizyjnego w charakterze supergwiazdora, wciel się w postać Szczura Rolanda.

Budzisz się i co się okazuje na budziku już 8.30, a przecież punktualnie o 9.00 masz swoje codzienne wystąpienie w telewizji. Nie możesz się spóźnić, rozpoczynasz więc swój wielki wyścig z czasem. Droga do celu nie jest prosta ani bezpieczna.

Poruszasz się w skomplikowanej sieci podziemnych kanałów, gdzie przez cały czas ganiają cię buty-mutanty zabierające przy spotkaniu niezbędną ci energię. Trzeba się bronić. Służy do tego pistolet z klejowymi pociskami, które zatrzymują buty, ale tylko na parę chwil. Jeżeli jesteś nieuważny i przypadkiem nadepniesz na niecelny pocisk — na chwilę sam się przykleisz, a przecież czas biegnie nieubłaganie. Co począć jeżeli w klejowym pistolecie zabraknie amunicji? (wskaźnik dolny). Patrz uważnie — może odnajdziesz kubły pełne kleju... Nie zapomnij też o górnym wskaźniku zapasu energii potrzebnej ci do życia i dalszej wędrówki. Uzupełnisz ją napotkanymi po drodze przysmakami. Ale uważaj — nie objadaj się nimi bez potrzeby. Nie możesz najeść się na zapas, a co będzie, gdy zabraknie ci pożywienia? Drzwi do studia telewizyjnego możesz otworzyć zagubionym gdzieś kluczem, jednak dopiero wtedy, gdy całe będą pokryte błyszczącym materiałem, którego kawałki rozrzucone są po kanałach. Musisz je wszystkie odszukać w labiryncie. Jednorazowo możesz udźwignąć tylko jeden taki kawałek, więc sieć kanałów przyjdzie ci przemierzać wielokrotnie. Aby przyśpieszyć swą wędrówkę możesz na dolnym poziomie skorzystać z metra, zatrzymując je na chwilę celnym strzałem ze swego klejowego pistoletu. Największym twoim wrogiem w tej grze jest czas. Bez dokładnej znajomości topografii kanałów i wytyczenia najkrótszych dróg ukończyć jej nie sposób. Załóżmy jednak, że do 9.00 zostało jeszcze parę minut. Drzwi do studia poryte są już w całości błyszczącą materią i Roland — czyli ty, graczu, biegniesz do nich z odnalezionym z trudem kluczem. Chyba wszystko skończy się dobrze i nie spóźnisz się na swój występ. Aby jednak uniknąć rozczarowań i zdenerwowania przed otworzeniem drzwi studia najedz się do syta, bo do wielkiego finału jest jeszcze daleko.

Jarosław Rumiak


TV
IBM
FIS
M


7.
8.
9.
10.
11.
12.
do 1.
7.
8.
9.
10.
11.
12.
TV
do 1.
Trasy przejść rurowych

CO JEST GRANE



Na nasze piąte notowanie nadeszło 1047 propozycji. Utworzyły one następującą dziesiątkę:

1. THREE WEEKS IN PARADISE !
2. SKOL DAZE !
3. SABOTUR !
4. KNIGHT LORE ▽
5. RESCUE ON FRACTALUS !
6. MATCH POINT △
7. ROLAND ON THE ROPES !
8. ROAD RACE !
9. MUGSY ▽
10. THE WAY OF THE EXPLODING FIST ▽

Przypominam wszystkim jeszcze raz zasady głosowania. Każdy nadsyła własną najlepszą jego zdaniem dziesiątkę gier. Wszystkie propozycje tworzą Bajtkową Listę Przebojów. Ten kto nadeśle taką samą dziesiątkę jaka ukaże się w Bajtku wygrywa, NAGRODĘ SPECJALNĄ. Jak dotąd nikt nie wytypował tej „złotej dziesiątki". Jest natomiast nagroda za najlepsze opisy typowanych gier. Prenumeratę Bajtka na rok 1986/87 otrzymuje Roman Graczyk z Bydgoszczy.

Sławek

BALLBLAZER



Dzisiaj w fantastycznym świecie komputerów zapanuje sport. Nie jest przypadkiem, że producenci gier sięgają także i po te dziedziny ludzkiego życia. Przyznajmy, że większość programów żąda od gracza przede wszystkim zręczności, refleksu i umiejętności koordynacji ruchów, czasami tylko wymaga wysilenia także szarych komórek. Sport jest zaś dziedziną, w której trzeba się wykazać wyżej wymienionymi cechami sprawnościowymi w szczególny sposób.

Gra, którą chcemy dziś zaprezentować łączy w sobie elementy piłki nożnej z ..., no powiedzmy, z bojerami. A wszystko zaczęło się dawno, dawno temu, gdy w zbadanym wszechświecie szalała krwawa wojna... To jednak należy do przeszłości. Dziś jest rok 3977 i „znajdujemy się w półprzestrzennym ogniwie, w podwójnym systemie gwiazd Kalaxon i Kalamar, w miejscu gdzie panuje zero grawitacyjne. Za moment, na powierzchni sztucznej planetoidy specjalnie zbudowanej na potrzeby sportu, rozpocznie się finał wielkiego turnieju INTERSTELLAR w ballblazer. Tym razem jest to największy w historii, w całym wszechświecie pojedynek Mistrzów. Po raz pierwszy do wielkiego finału zakwalifikowała się istota z planety Ziemia. Walczyła w niezliczonych rundach kwalifikacyjnych, wytrwała i zatryumfowała wśród bezmiarów kosmosu zdobywając prawo do wzięcia udziału w turnieju o honor swej planety i ostateczny tytuł, którego nie można już utracić — MASTERBLAZER."

Ballblazer to nie tylko sport. To sprawdzian i forsowny trening dla układu nerwowego. Kiedy komputer pokładowy obróci cię o 90 stopni, właśnie utrzymanie przytomności umysłu i spokoju może zadecydować o wszystkim. Ale warto... jeśli przejdziesz zwycięsko przez próby, przed którymi stawia cię ballblazer, to później wojna z całą galaktyką wyda ci się zabawą.

Ballblazer zawiera także elementy ćwiczeń wojskowych. Używa się pojazdów zwanych Rotofil'ami, wyposażonych w komputer pokładowy, który automatycznie obraca pojazd. Ty jesteś we wnętrzu takiego Rotofila, podobnie jak twój przeciwnik. Ekran jest podzielony tak, że każdy z graczy posiada widok ze swojego Rotofila na przebieg gry — ty widzisz Rotofil przeciwnika i odwrotnie.

Wyobraź sobie, że twój Rotofil jest otoczony przez dużą poduszkę energii — jest to twoje pole uderzeniowe. Kiedy piłka plazmowa zbliży się na odpowiednią odległość twoje pole uaktywnia się. Moment ten sygnalizuje ci odpowiednie buczenie głośnika.

Gdy piłka plazmowa zostaje wystrzelona wystartuj do niej i przechwyć ją, zanim zrobi to twój przeciwnik. Bramka porusza się w tym samym kierunku, w którym została wystrzelona piłka. Odnajdź ją i postaraj się wstrzelić piłkę między słupki. W zależności od tego z jakiej odległości strzelisz gola przyznawana jest odpowiednia ilość punktów. Najwyżej punktowane są strzały dalekosiężne. Aby wygrać trzeba zdobyć maksymalną ilość punktów — jest ich pięć.

W momencie przechwycenia piłki w twoje pole siłowe, Rotofil automatycznie ustawia się przodem do bramki. Usłyszysz wtedy dźwięk i zobaczysz zmianę koloru piłki. Jeżeli piłka znajduję się w posiadaniu przeciwnika możesz mu ją zabrać podchodząc do niego z boku i wystrzeliwując piłkę z jego pola energetycznego.

Pamiętaj, że prawdziwy gracz nigdy nie traci orientacji podczas nagłych zwrotów.

W całym wszechświecie honorowany jest tylko trzyminutowy ballblazer, dwóch graczy i jeden zwycięzca. Stać się graczem jest łatwo ale mistrzem bardzo trudno. Ale zrób pierwszy krok i spróbuj odnaleźć siebie.

(ap)

MATCH POINT



Jeżeli lubisz grać w tenisa a na dworze pada właśnie deszcz, zagraj w Match Point. Jest to najlepsza sportowa gra na Spectrum. Ma dokładną, staranną grafikę i zbliżone do naturalnych efekty dźwiękowe. Można grać z komputerem, z drugim graczem lub przyglądać się grze w programie demonstracyjnym. Gra odbywa się na poziomie ćwierć-, pół-, i finałów. W ćwierćfinałach jest powolna i mało dynamiczna. Na poziomie półfinałów można już uzyskać przyspieszenie piłki: przy serwie jest to dół+strzał, w grze strzał+którykolwiek z klawiszy kierunkowych (lewo, prawo, góra lub dół). W finałach wszystkie piłki są bardzo szybkie.

Mecze rozgrywać można po jednym, trzech lub pięciu setach. Program pozwala na wprowadzenie imion lub pseudonimów graczy i wyświetla je na specjalnych tablicach świetlnych.

Punktacja odbywa się zgodnie z regulaminem mistrzostw świata w tenisie ziemnym. Mamy też piłki meczowe i setowe, na stołeczku siedzi sędzia w krótkich spodenkach, a obok niego chłopcy do podawania piłek. Wszystko jest w prawdziwym meczu. Spróbuj być lepszy od Fibaka! Gra joystickiem Kempston, Sinclair lub z klawiatury. Wygrywający dostaje piękny wielki puchar. Powodzenia!

(mp)

MIKROKOMPUTER WSPÓŁAUTOREM GRY

O tym, że komputer może być partnerem w grze, wiedzą nie tylko czytelnicy „Bajtka". Dzisiaj pokażemy komputer w innej roli — współprojektanta gry planszowej.

Autorzy otrzymali zamówienie na opracowanie gry planszowej dla dzieci. Producent postawił następujące wymagania:

— tematem gry ma być wyścig samochodowy,
— gra powinna mieć stosunkowo proste przepisy i rekwizyty — przeznaczona jest dla dzieci od lat 6,
— nie powinna być to gra losowa, lecz strategiczno-losowa.

Gier spełniających dwa pierwsze warunki opracowano już sporo. Należało zatem zaproponować jakiś oryginalny element, który czyniłby naszą grę „inteligentną" i atrakcyjną. Tak więc o ruchach samochodów nie decyduje jedna kostka ale trzy różne:

— czerwona — standardowa — o liczbach oczek: 1, 2, 3, 4, 5, 6,
— niebieska — „o małej wariancji" o liczbach oczek: 3, 3, 3, 4, 4, 4,
— żółta — „o dużej wariancji" o liczbach oczek: 0, 0, 0, 4, 8, 9.

Gra polega oczywiście na przejechaniu toru ze szczęśliwymi i pechowymi przygodami. Samochody zdążając do mety poruszają się o wyrzuconą liczbę oczek. Ale przed każdym ruchem gracz decyduje, którą kostką wykona rzut — może więc oddziaływać na los przez wybór rozkładu prawdopodobieństwa. Jak widać, wartość oczekiwana (średnia arytmetyczna) liczby oczek na każdej z trzech kostek jest taka sama. Różna jest natomiast wariancja (rozrzut w stosunku do wartości oczekiwanej). Można przypuszczać, że gracz ostrożny będzie rzucał najczęściej kostką niebieską, ryzykant — kostką żółtą. Autorzy uznali, że myślenie powinno być w grze premiowane. Rozsądny gracz winien mieć większe szanse na zwycięstwo. I tu właśnie pojawiła się możliwość wykorzystania komputera. Należało tak rozmieścić na torze przygody (pola korzystne i niekorzystne), aby gracz rozsądny zyskiwał przewagę nad graczami rzucającymi wciąż tą samą kostką. Można to było zrobić metodą tradycyjną tzn. rozgrywając setki wyścigów na różnych torach. Opracowane statystycznie wyniki takiego eksperymentu pozwoliłyby wybrać odpowiedni tor, preferujący gracza maksymalizującego prawdopodobieństwo wejścia na pola korzystne i minimalizującego szanse trafienia na niekorzystne w stosunku do graczy wybierających kostkę losowo, bądź rzucających uparcie jedną kostką. Jak widać, metoda tradycyjna wymaga dużo czasu i cierpliwości.

Znacznie szybciej rozwiązano problem stosując symulację komputerową. Przedstawiony poniżej program na ZX Spectrum symuluje 10000 rzutów kostką (liczbę rzutów określa zmienna rmax) wybraną według określonej strategii (linie 210-260). Instrukcje zawarte w liniach 100-150 służą do wczytania toru. Użyte tu symbole oznaczają:

— „n", „c", „z" — pola, na których gracz musi rzucać odpowiednio niebieską, czerwoną lub żółtą kostką,
— „w" — wypadek — strata jednej kolejki,
— „r" — premia: dodatkowy rzut kostką,
— „t" — premia: przesunięcie o 3 pola do przodu,
— „ " — pole neutralne (nic się nie dzieje).

Wybrany do produkcji tor składa się z 56 pól zakodowanych w instrukcji DATA. Aktualne położenie samochodu na torze określa zmienna t, liczbę przejechanych okrążeń zmienna lokr, a liczbę wykonanych rzutów zmienna lr. Liczba oczek na kostce generowana jest przez procedury symulujące kostki (1000 — żółtą, 2000 — niebieską, 3000 — czerwoną).

Jeden eksperyment symulujący 10000 rzutów według wybranej strategii trwał ok. 25 minut. Tak więc w ciągu jednego dnia można było zbadać co najmniej 5 torów, po 5 strategii na każdym. Rezultaty uzyskane dla ostatecznej wersji toru zamieszczone są w tabeli.

Strategia wyboru kostki	Liczba okrążeń
kostka żółta	623.21
kostka niebieska	624.18
kostka czerwona	620.23
wybór losowy	622.12
wybór „rozsądny"	673.50

Na podstawie wyników symulacji uznaliśmy, że wybrany tor spełnia nasze postulaty. Można przypuszczać, że istnieje taki rozkład pól na planszy, który w jeszcze większym stopniu preferuje gracza rozsądnego. Przy wyborze toru należy jednak brać pod uwagę również walory estetyczne (wygląd planszy), w czym człowiek jest jak dotąd niezastąpiony.

Zgodność wyników badania symulacyjnego z praktyką można porównać grając w „MINI GRAND PRIX". Życzymy przyjemnej zabawy.

Jan Rojewski
Michał Stajszczak

```
10 INPUT "rmax = "; rmax
100 INPUT "tor = "; tor
110 DIM t$(tor,1)
120  FOR i=1 TO tor
130  READ t$(i,1)
140  NEXT i
150 DATA " "," "," ","z"," "," "," ","t","
 "," "," ","w"," "," "," ","t"," ","n"," ","
 ","w"," ","c"," ","r"," "," "," ","t"," ",
"z"," "," "," "," ","w"," ","c"," ","w"," "
," ","r"," "," "," ","w","z"," "," "," ","
 ","n"," "," "," "
200 DIM s(tor)
210 PRINT "Rodzaj strategii:"'"1 - stale ko
stka zolta"'"2 - stale kostka niebieska"'"3
 - stale kostka czerwona"'"4 - losowy wybor
 kostki"'"5 - strategia rozsadna"
220 INPUT "STRATEGIA ";st
230 IF st<5 THEN FOR i=1 TO tor: LET s(i)=s
t: NEXT i: GOTO 270
240  FOR i=1 TO tor
250  INPUT ("s(";i;") = "); s(i)
260  NEXT i
270 LET lr=0
280 LET t=tor: LET lokr=-1
290 GOTO 390
300 LET t=t+INT rzut
310 IF t>tor THEN LET t=t-tor: LET lokr=lok
r+1
320 IF rzut=0 THEN GOTO 390
330 IF t$(t)="t" THEN LET t=t+3: GOTO 390
340 IF t$(t)="r" THEN LET lr=lr-1: GOTO 390
350 IF t$(t)="w" THEN LET lr=lr+1: GOTO 390
360 IF t$(t)="z" THEN LET nast=1: GOTO 400
370 IF t$(t)="n" THEN LET nast=2: GOTO 400
380 IF t$(t)="c" THEN LET nast=3: GOTO 400
390 LET nast=s(t)
400 GOSUB 1000*nast
410 LET lr=lr+1
420 IF lr>rmax THEN GOTO 500
430 GOTO 300
500 PRINT lokr,t
600 STOP
1000 LET x=6*RND
1010 IF x<3 THEN LET rzut=0: RETURN
1020 IF x<4 THEN LET rzut=4: RETURN
1030 IF x<5 THEN LET rzut=8: RETURN
1040 LET rzut=9: RETURN
2000 LET rzut=2*RND+3
2010 RETURN
3000 LET rzut=6*RND+1
3010 RETURN
4000 LET x=INT (3*RND)+1
4010 GOSUB 1000*x
4020 RETURN
```

KONKURS
NA NAJLEPSZĄ STRATEGIĘ W „MINI GRAND PRIX"

Firma „DAS" (producent „MINI GRAND PRIX") ufundowała nagrody dla autorów najlepszych strategii tzn. takich, które zapewniają przejechanie najdłuższej trasy przy wykonaniu 10000 rzutów na torze zakodowanym w instrukcji 150. Rozwiązanie powinno mieć następującą formę: np. 1-c, 2-n, 3-z tzn. na polu 1 należy rzucać czerwoną kostką, na 2 niebieską itd.

Rozwiązania należy nadsyłać na adres: DAS, skr. poczt. 1, 05-204 Okuniew k/Warszawy w terminie do 1 grudnia 1986 r.

PORADNIK MŁODEGO PIRATA cz. I



Poniższy artykuł (a właściwie pierwszą jego część) przeznaczamy dla tych wszystkich, którzy myślą o ekonomicznym wykorzystaniu swoich dyskietek zaśmieconych programami działającymi równie dobrze z taśmy. Ponadto autorzy opisują jak uczynić „nieprzegrywalne" przegrywalnym — czyli po prostu jak kopiować programy dyskowe o długości do 207 bloków.

Wielokrotnie mieliśmy już okazję spotkać świeżo upieczonych posiadaczy Commodore 64 łamiących sobie głowę nad opracowaniem złotego sposobu umożliwiającego przegrywanie programów dyskowych na taśmę. Najczęściej oczywiście chodziło o tak renomowane gry jak „Kennedy Approach", „Summer Games" czy też „Silent Service", rzadziej zaś o jednoczęściowe programy mające po 200 i więcej bloków (1 blok — 256 bajtów). Znajdowali się również chętni do przegrywania programów kilkuczęściowych, wgrywanych kolejno do pamięci za pomocą krótkiego programu wczytującego, tzw. loadera.

Większość zainteresowanych tym problemem osób kopiowała programy za pomocą znanego programu „COPY 190" (kopiującego programy o długości do 190 bloków, dającego jednakże LOAD ERROR chociaż program działa), albo poprzez najróżniejszej maści i formatu programy SUPEREXTRAHIPERCUDOCOPY, z reguły działające wprost odwrotnie niż po ich nazwach można by się było tego spodziewać. Bardzo rzadko udawało nam się spotkać kogoś, kto starałby się rozwiązać ów problem metodami naturalnymi czyli bez pośrednictwa wyżej wymienionych programów kopiujących. Z tego też powodu chcielibyśmy przedstawić Czytelnikom parę pomysłów i rozwiązań umożliwiających przegrywanie NIEKTÓRYCH programów dyskowych na taśmę w systemie przyspieszonego zapisu i odczytu TURBO. Dla niewtajemniczonych podajemy, że system ten pozwala na dziesięciokrotne przyspieszenie zapisu, odczytu oraz weryfikacji programów taśmowych.

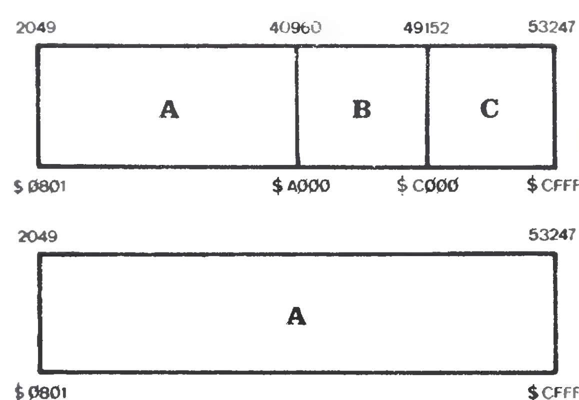
W tym miejscu chcielibyśmy jednak ostudzić rozgrzane głowy wszystkich „piratów" — amatorów marzących o posiadaniu na taśmie wszystkich absolutnie programów dyskowych. Prezentowane przez nas sposoby dotyczą (jak to już wyżej zaznaczyliśmy!!!) tylko NIEKTÓRYCH programów, z reguły zresztą tych „gorszych". Przegrywanie „Kennedy Approach" czy „Summer Games" wymaga przede wszystkim doskonałej znajomości zasad programowania w języku maszynowym i systemu operacyjnego nie mówiąc już o rozeznaniu w technikach zabezpieczania programów czy olbrzymiej ilości czasu oraz cierpliwości. Rozsądniejszym proponujemy po prostu poczekać — z reguły większość tych „lepszych" programów ukazuje się najpierw w wersji dyskowej by po jakimś czasie znaleźć się również na firmowych kasetach. Z tymi zaś potrafi już sobie dać radę nawet przedszkolak...

Drugim kubłem jeszcze zimniejszej wody będzie spora dawka teorii o której niestety trzeba coś wiedzieć zanim przystąpi się do przegrywania. Teorię tę trzeba także ZROZUMIEĆ, gdyż w przeciwnym wypadku szkoda tylko Twojego czasu na bezowocne próby przegrywania.

Ze względu na obfitość materiału nasz artykuł podzieliliśmy na trzy części. Postaramy się w nich

omówić w miarę dokładnie i szczegółowo wszystkie te zagadnienia, których znajomość jest niezbędna do skutecznego przegrywania programów dyskowych (tylko NIEKTÓRYCH!) na taśmę. Zacznijmy więc od teorii...

Czy zastanawiałeś się kiedykolwiek drogi Czytelniku jaka jest różnica pomiędzy LOAD „PROGRAM", 1 i LOAD „PROGRAM", 1,1? Jeżeli nie to postaramy Ci się różnicę tę w miarę dokładnie wytłumaczyć, gdyż jest ona w istocie kolosalna. Pamięć RAM (Random Access Memory — pamięć o swobodnym dostępie) dla użytkownika zaczyna się w Commodore 64 od adresu dziesiętnego 2048 (i w normalnej konfiguracji kończy w komórce o adresie 40959 ($ 9FFF). Ponieważ komórka 2048 zawiera zawsze wartość 0 dla naszych celów przyjmiemy nieco inne ustawienie — od adresu 2049 ($ 0801) Następne 8 kb od adresu 40960 do 49151 ($ A000–$ BFFF) to obszar „maskowany" przez interpreter języka BASIC — jest to pamięć dla nas niedostępna dopóki nie zostanie on wyłączony. Poprzez „maskowanie" rozumiemy tu pewien kruczek techniczny, który pokrótce postaramy się objaśnić. Otóż cały interpreter jest w rzeczywistości zawarty w pamięci ROM (Read Only Memory) czyli pamięci, którą możemy tylko „czytać" — nie możemy zaś do niej nic wpisać. W zależności od stanu mikroprocesora (ustawienia go na „czytanie" bądź „zapisywanie" danych do pamięci) możemy co prawda wpisać dane do obszaru pamięci leżącej „pod" interpreterem jednakże ze względu na konstrukcję Commodore 64 odczytać możemy jedynie interpreter. Innymi słowy chociaż wpisujemy dane do RAM odczytać możemy



Rys.1. Konfiguracja pamięci przed wyłączeniem interpretera BASIC (a) oraz po jego wyłączeniu (b)
A — Pamięć RAM dostępna dla użytkownika
B — Obszar pamięci „maskowany" przez interpreter
C — Dodatkowe 4 kb pamięci RAM nieadresowanej przez BASIC

jedynie ROM. Stan ten będzie trwał dopóki interpreter nie zostanie wyłączony. Odczytywanie danych jednakże będzie możliwe jedynie w języku maszynowym, gdyż po wyłączeniu interpretera BASIC nie działa.

Taki stan rzeczy powinien również wyjaśnić Ci jakim cudem jest możliwe wczytanie do pamięci programu o długości np. 45 kb pomimo, że w normalnym układzie dostępne dla Ciebie jest łącznie ok. 43 kilobajtów pamięci. Podczas wczytywania programów mikroprocesor jest ustawiony na „zapis" danych do pamięci, może więc dane te wpisać do RAM leżącej pod interpreterem (adresy 40960–49151, $ A000–BFFF). Skorzystanie z nich będzie jednakże możliwe dopiero po wyłączeniu interpretera, gdyż ustawienie mikroprocesora na „czytanie" spowoduje wyczytywanie danych z ROM. W praktyce wyłączanie go powoduje krótka, kilkunastobajtowa procedura umieszczona na początku programu, lecz dopiero po jego uruchomieniu, gdyż będzie on nam potrzebny do rozpoznania i wykonania instrukcji SYS uruchamiającej nasz program.

Po interpreterze i leżącej pod nim RAM do dyspozycji użytkownika pozostaje jeszcze 4 kb pamięci RAM od adresu 49152 do 53247 ($ C000–$ CFFF). Obszar ten jest nieadresowalny przez' BASIC (nie można w tym obszarze — w normalnej konfiguracji — umieścić programu napisanego w BASIC-u). Co to wszystko właściwie oznacza? Oznacza to, że BASIC może „obsłużyć" program jedynie w obszarze od adresu 2049 do 40960, w pozostałych zaś 4 kb można jedynie umieścić np. dane dla sprite'ów czy procesora muzycznego. Wyrażenie „normalna konfiguracja" oznacza, że początek pamięci RAM dla użytkownika zaczyna się w komórce o adresie 2049 i kończy w 40959. W rzeczywistości konfiguracja ta może być zmieniana w dowolny sposób, co będzie miało duże znaczenie przy omawianym przez nas temacie.

Jak Ci zapewne wiadomo, program napisany w BASIC może mieć co najwyżej 38911 bajtów długości, zaś program maszynowy nawet ponad 52 kb. Skąd ta różnica? Właśnie z wyłączonego interpretera. Po jego wyłączeniu programista zyskuje dodatkowe 8 kb pamięci co sprawia, że ma on do dyspozycji jeden ciągły blok pamięci RAM od adresu 2049 ($ 0801) do 53247 ($ CFFF). Oczywiście obszar ten będzie musiał być obsługiwany przez język maszynowy gdyż wyłączenie interpretera powoduje wyłączenie BASIC.

Co do rekonfigurowania pamięci Commodore 64 to przy minimum wprawy można dojść do zaskakujących rezultatów. Pomijając możliwość operowania dwoma całkowicie różnymi programami umieszczonymi naraz w pamięci, to można również wspomniane wyżej obszary pamięci zamieniać ze sobą. Dobrym przykładem może tu być krótki program napisany w BASIC wykorzystujący np. 35 kb pamięci do przechowywania danych dla procesora muzycznego. W takim wypadku możliwe jest umieszczenie głównego programu np. w obszarze pamięci od 49152 do 53247 ($ CFFF) i danych do niego w adresach 2049–40959 ($ 0801–$ 9FFF). Rekonfiguracji takiej można dokonać za pomocą czterech POKE'ów! Ale o tym za miesiąc.

KD i MS

Materiały do Klan Commodore przygotowali: Klaudiusz Dybowski, Michał Silski.

PERKUSJA NA C64

```
100 S=54272.FORA=STOS+24:POKEA,0.NEXT
110 POKES+5,5:POKES+6,5:POKES+24,15
120 P=16:E=128:DIMR(P-1),B(3)
130 B(0)=0:B(1)=20:B(2)=80:B(3)=250
140 POKE53280,0.POKE53281,0
150 PRINTCHR$(147),CHR$(30)
160 PRINT"NUMERY DOSTEPNYCH BRZMIEN: 0,1
,2,3"
170 PRINT
180 FORA=0TOP-1:PRINTR(A);CHR$(157);:NEX
T:PRINT
190 PRINT
200 PRINT"F1 - ODTWARZANIE, F3 - WPROWAD
ZANIE"
210 GETA$:IFA$<CHR$(133)ORA$>CHR$(134)TH
EN210
220 ONASC(A$)-132GOTO300,400
300 FORA=0TOP-1
310 POKES+1,B(R(A)):POKES+4,E+1
320 POKE1145+2*A,(R(A)+48)OR128
330 FORD=0TOSZ:NEXT
340 POKES+4,E
350 POKE1145+2*A,R(A)+48
360 NEXT
370 GETA$:IFA$<>""THEN210
380 GOTO300
400 FORA=0TOP-1
410 POKE214,11:PRINT:PRINT"     ",R(A),CHR$
(157);CHR$(157);CHR$(157);CHR$(157);
420 INPUTR(A).IFR(A)<0ORR(A)>3THEN410
430 POKE1145+2*A,(R(A)+48)OR128
440 NEXT
450 INPUT"SZYBKOSC";SZ
460 GOTO210
READY.
```

Zdarza się czasem, że posiadacz mikrokomputera nie jest do reszty stracony dla świata i interesuje go coś jeszcze poza grzebaniem się w bajtach lub ustanawianiem rekordów w ilości zestrzelonych statków kosmicznych.

Poniższy program przeznaczony jest dla tych, których zainteresowania mają coś wspólnego z muzyką. Co prawda opublikowana wersja raczej nie zainteresuje profesjonalistów, ale wielu amatorów być może do czegoś to wykorzysta. Zresztą, programowanie i odtwarzanie własnych rytmów za pomocą własnoręcznie „wklepanego" i (mam nadzieję) rozbudowanego programu może być po prostu dobrą zabawą.

Po wpisaniu, program należy zapisać na taśmie lub dysku. Zasadą jest, by robić to przed uruchomieniem, ponieważ błąd w programie może spowodować „zawieszenie" komputera i konieczność ponownego wpisywania. Mając zapisaną kopię wystarczy ją załadować i poprawić błąd.

Jeżeli udało się już szczęśliwie program uruchomić, można przystąpić do wprowadzania własnych rytmów. Zasada jest taka sama, jak w „normalnych" programowanych perkusjach elektronicznych. Pojedyńczy program perkusji składa się z określonej liczby „miejsc na uderzenia". W naszym przypadku miejsc tych jest 16. W każde takie miejsce można wpisać dowolny rodzaj uderzenia bądź pauzę (ciszę). W opisanym programie wprowadzanie odbywa się przez podawanie liczb w zakresie 0÷3, gdzie 0 to pauza a 1, 2, i 3 to uderzenia o coraz większej wysokości. Naciśnięcie klawisza RETURN bez podania liczby powoduje pozostawienie w pamięci starej wartości. Po wprowadzeniu szesnastu cyfr pojawia się pytanie o szybkość. Należy wpisać wartość z zakresu 0 — 200. Im wyższa liczba, tym wolniejszy rytm.

Sterowanie funkcjami programu odbywa się za pomocą klawiszy F1 i F3. Program może znajdować się w trzech stanach: czekania, wprowadzania i odtwarzania. Stan odtwarzania wybieramy naciskając klawisz F1. Odtwarzanie można przerwać dowolnym klawiszem (byle nie STOP, bo to zatrzyma cały program). Wprowadzanie uruchamia się klawiszem F3. Po zakończeniu wpisywania program przechodzi w stan oczekiwania. Można wtedy posłuchać zaprogramowanego rytmu (F1) bądź ponownie go zmienić (F3).

Polecam zanalizowanie działania programu. Ponieważ najistotniejsze parametry programu zostały zdefiniowane jako zmienne w liniach 120 — 130, wszelkie eksperymenty są łatwe do wykonania.

A oto opis ważniejszych parametrów: tablice R i B zapamiętują odpowiednio numery brzmień każdego uderzenia i odpowiadające tym numerom wartości. Zmienna P ustala ilość uderzeń przypadającą na pojedyńczy rytm (podział rytmiczny). Zmienna E ustala rodzaj fali, która ma być wytworzona przez syntezator. Wartość 128 to szum; 16, 32 i 64 to dźwięki o różnych brzmieniach. Jeśli E będzie miała wartość inną niż 128 (czyli generowany będzie dźwięk o konkretnych wysokościach), to odpowiednio ustalając wartości tablicy B można „zmusić" perkusję do grania melodyjek, a więc przerobić ją na sekwencer.

Poniżej podane są przykłady rytmów, możliwych do osiągnięcia za pomocą opisanej perkusji:

1 1 0 0.2 0 0 1.0 0 1 0.0 0 2 0, szyb. 80
1 1 1 0.3 0 0 1.1 0 1 0.3 0 2 2, szyb. 40
1 0 2 1.0 1 2 0.1 1 2 0.1 1 2 0, szyb. 60
3 3 3 3.2 2 2 2.1 1 1 1.0 0 0 0, szyb. 30

Kropki między cyframi zostały wstawione w celu ułatwienia ewentualnego wpisywania.

Na zakończenie jeszcze jedna uwaga: perkusja elektroniczna, opisana tutaj, powinna posłużyć wszystkim zainteresowanym jako wzór bądź pomysł do dalszej pracy. Rozbudowując program, dodając możliwość programowania wielu rytmów jednocześnie, wzbogacając brzmienia lub dodając możliwość sterowania urządzeniami zewnętrznymi (najwygodniej za pomocą tzw. USER PORT, czyli równoległego portu WE/WY) można dojść do stworzenia wartościowego programu muzycznego, mogącego zaspokoić nawet najbardziej wymagających. Oczywiście nie obejdzie się bez języka maszynowego, bowiem BASIC, jak na potrzeby muzyki, jest stanowczo zbyt powolny i nie zapewnia właściwej stabilności czasowej. Życzę wesołej zabawy.

KORESPONDENCJA Z FRANCJI

ICH PIERWSZY KOMPUTER!

Wybaczcie Drodzy Czytelnicy, ale nie dane mi było szukać komputerowych centrów handlowych na paryskim bruku. Dokładnie przeto spenetrowałem każdy sklep oferujący elektroniczne cacka w położonym nad Kanałem La Manche prowincjonalnym Boulogne, gdzie przyszło mi spędzić dwutygodniowy urlop.

Naprzód mała refleksja. Nawet w niewielkich miescinach oddalonych bardzo od metropolii nad Sekwaną mikroelektronika obecna jest wszędzie. Nie chcę odkrywać Ameryki pisząc, że w najmniejszych nawet sklepach używa się elektronicznych kas i wag błyskawicznie kalkulujących należność za towary sypkie, że wiejskie biuro podróży posiada końcówkę z dostępem do wielkich stołecznych, a nawet zagranicznych terminali napchanych potrzebnymi informacjami. Ale warto chyba zauważyć, że podobne urządzenia zainstalowane są w prywatnych przewoźnych straganach zbudowanych na podwoziach specjalnych samochodów wędrujących wraz z właścicielami po placach targowych północnej Francji.

A teraz do sklepów. Jak mi wytłumaczono, poza stolicą nie bardzo opłaca się trzymać w sklepie wyłącznie komputery. Mikroelektronika (maszyny do pisania, kalkulatory, mikroprocesory), jest więc oferowana na równi z papeteriami i materiałami biurowymi, z widokówkami, płytami, kasetami magnetowidowymi, maszynami do kopiowania i najzwyklejszymi na świecie flamastrami. Oczywiście wszystko, co francuskie... musi być najlepsze i stąd polityka potężnych ceł na wszystko, co w sklepach Kraju Marianny pochodzi z importu, a może stanowić wyzwanie dla rodzimej produkcji. Ten sam schemat dotyczy również mikrokomputerów.

Najtańsze, najlepiej reklamowane i eksponowane są mikroprocesory francuskiej firmy elektronicznej Thomson (wytwarza praktycznie wszystko, od zegarkowych kalkulatorów do systemów naprowadzania pocisków balistycznych). Spectrum Sinclaire'a nie widziałem, a klawiaturę Commodore'a dostrzegłem tylko w jednym sklepie.

Z reguły nabywcy proponuje się komplet. Na najprostszy zestaw Thomsona składa się komputer MO 5 (porównywalny ze Spectrum), magnetofon i dwie kasety z programami gier. Całość (specjalna oferta) 2990 franków (dolar wówczas oscylował w granicach 6,7 franka). Obdarzony większą pamięcią TO 7 z identycznym, co jego poprzednik wyposażeniem — 3990 franków.

64 kilobajtowy Schneider CPC 464 z monitorem GT 65 oferowany jest w dwóch wydaniach. Ekran w kolorze zielonym — 2690 franków, ekran kolorowy — 3990.

Amstrad PCW 8256 z drukarką i monitorem kosztuje 5926 franków. Amstrad CPC 6128 z kolorowym monitorem i magnetofonem wystawiony na okazyjnej wyprzedaży — 5990 franków (tyle co nowiutki magnetowid).

Informacja dla amatorów zabawy przy migającym ekranie — kaseta z oprogramowaniem do gier kosztuje od 120 do 180 franków. Urządzenie ATARI 2600 przeznaczone wyłącznie do gier jest stosunkowo tanie — 699 franków. Każda gra na nośniku tej firmy — 112 franków.

Profesjonalne komputery użyteczne w handlu i księgowości, z pamięcią od 500 kilobajtów w górę, przypominające wyroby spółek Apple i IBM, ze stacją dyskie-

---

KORESPONDENCJA Z WĘGIER

BAJTEK NA UNIWERSYTECIE

Na początku lipca tego roku, w Węgierskim mieście Salgotarian młodzi ludzie z wielu krajów spotkali się na Letnim Uniwersytecie. Tego rodzaju spotkania były organizowane dla węgierskiej młodzieży już od wielu lat, po raz trzeci natomiast zaproszono młodzież z innych krajów. Temat-hasło tegorocznego Uniwersytetu to „Technologie przyszłości i młodzież".

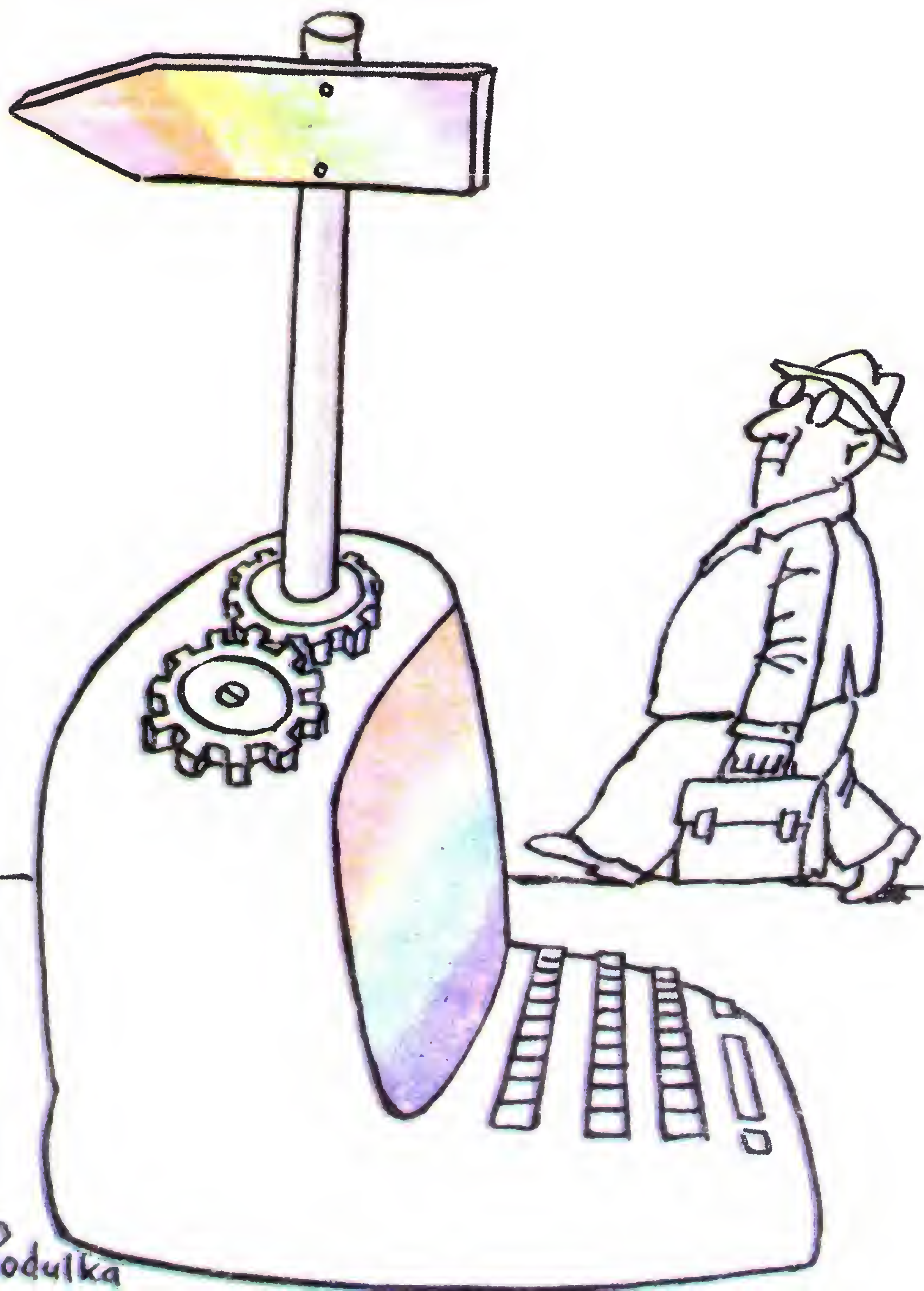
„Bajtek" otrzymał zaproszenie do udziału w zajęciach tej niezwykłej Alma Mater i — mimo, że wakacje służą raczej odpoczynkowi od nauki — przyjął je z radością. Nie codzień zdarza się bowiem okazja spotkać i porozmawiać na interesujące nas tematy jednocześnie z młodymi ludźmi z trzynastu krajów. Byli więc Rosjanie i Amerykanie, Szwedzi, Francuzi a także Bułgarzy i Czechosłowacy. Przyjechali Niemcy z NRD i RFN, Austriacy oraz Szwaicarzy i Włosi. Rzecz jasna, że nie zabrakło gospodarzy — Węgrów.

Pomimo wspaniałej pogody i kuszącej perspektywy kąpieli w basenie, frekwencja na wykładach była praktycznie stuprocentowa. Z pewnością największe znaczenie miał w tym przypadku dobór tematów Oto kilka z nich: „Człowiek i komputer". „Nauka — technologia — edukacja". „Szkoła w roku 2000", „Rewolucja i przyszłość informacji". Znacznie dłużej niż same wykłady trwały dyskusje. Spowodowało to nawet pewne kłopty organizatorom, gdyż zajęcia nie mogły się nigdy zakończyć w przewidzianym czasie. Jednak żaden z uczestników nie skarżył się jedząc obiad godzinę czy dwie później niż było to planowane.

Dyskusje z sali wykładowej przenosiły się do domków i pokojów, na basen i wszędzie tam gdzie zebrało się kilkoro dziewcząt i chłopców, chcących się dowiedzieć jakie są poglądy ich przyjaciół z innych krajów. I choć w tych warunkach rozmowy bywały trudniejsze — trzeba się było obejść bez tłumacza a często słychać było jednocześnie kilka języków — nikt się tym nie zrażał.

Żegnając w ostatnim dniu moich nowych przyjaciół zwróciłem uwagę na dziwną rzecz: nikt z nas nie zdążył się nawet porządnie opalić. Aż trudno uwierzyć, że spędziliśmy razem dziesięć słonecznych, wakacyjnych dni we wspaniałym ośrodku wypoczynkowym na Węgrzech. Ale myślę, że nikt z nas nie żałował straconej opalenizny. Wróciliśmy z czymś więcej, z czymś co nie zniknie tak szybko — z wiedzą o innych krajach, o poglądach żyjących tam ludzi, o ich nadziejach i problemach. Tej wiedzy nie można zdobyć oglądając telewizję i czytając gazety.

Na koniec chciałbym się podzielić kilkoma spostrzeżeniami na temat chyba najbardziej interesujący naszych Czytelników — mikrokomputery na Węgrzech. Trzeba przyznać, że w tym przypadku nasi bratankowie wykazali się lepszym refleksem W chwili, gdy u nas trwała batalia o „Bajtka" i wielu ludzi nie wierzyło jeszcze, że wydawanie pisma mikrokomputerowego jest w ogóle celowe i opłacalne w naszym kraju. Węgrzy mieli już za sobą telewizyjny kurs BASIC-a. Kurs ten nie tylko umożliwiał zdobycie wiedzy z zakresu programowania ale również — po zdaniu odpowiednich egzaminów — dawał uprawnienia zawodowe Podobnie wygląda sytuacja w węgierskim szkolnictwie. Bardzo wiele szkół średnich podstawowych posiada już świetnie wyposażone pracownie komputerowe. Młodzież i dzieci uczą się posługiwania komputerem a także korzystają z jego usług w nauce innych przedmiotów.



tek i ekranem są już niesłychanie drogie, jak na kieszeń przeciętnego Francuza dysponującego pensją rzędu 5-7 tys. franków. System francuski GOUPI (512 kilobajtów) kosztuje 26850 franków plus 18,6 proc. podatku. Doskonalszy OMEGA-OCIETS z kolorowym monitorem — powyżej 50 000 franków.

Najbardziej jednak zafrapowało mnie coś, co dostrzegłem na wystawie konwencjonalnego sklepu z zabawkami. Rzecz cała, niesłychanie kolorowo wykończona, mieści się w fascynującym oko dziecka pudełku o wymiarach 30 cm × 40 cm × 10 cm, opatrzonym napisem „Mój pierwszy komputer". Po wyjęciu mamy do czynienia ze złożonymi do środka, przymocowanymi do siebie miniaturowym ekranem oraz równie małą (przystosowaną do dziecięcych palców) klawiaturką. Dostajemy do tego 6 programów zarejestrowanych na elastycznych dyskach o wymiarach kart do gry. Wprowadza się je do urządzenia przy pomocy otworu imitującego podobny w „zupełnie dorosłej" stacji dyskietek „poważnego komputera". 7–8-letni adept informatyki może bawić się, grać, a nawet porozumieć ze swym pierwszym w życiu komputerem. Całość, moim zdaniem niesłychanie ważna w całym cyklu mikroelektronicznej edukacji społeczeństwa, z uwagi na metkę „Made in France" kosztuje tylko 591 franków.

Wojciech Łuczak

O ile w Polsce trudno jest jednoznacznie określić, który z mikrokomputerów zdobył sobie największą popularność, to na Węgrzech bez wahania można powiedzieć, że jest to Commodore. Z tą firmą podpisano najwięcej umów, te mikrokomputery sprowadzono dla szkół. Węgrzy nie czekali z rozpoczęciem programu edukacji informatycznej na uruchomienie masowej produkcji własnych mikrokomputerów, postawili na początek na Commodore. Ma to rzecz jasna swoje wady zarówno gospodarcze jak i organizacyjne, ale w ten sposób znaleźli się jednak z przodu.

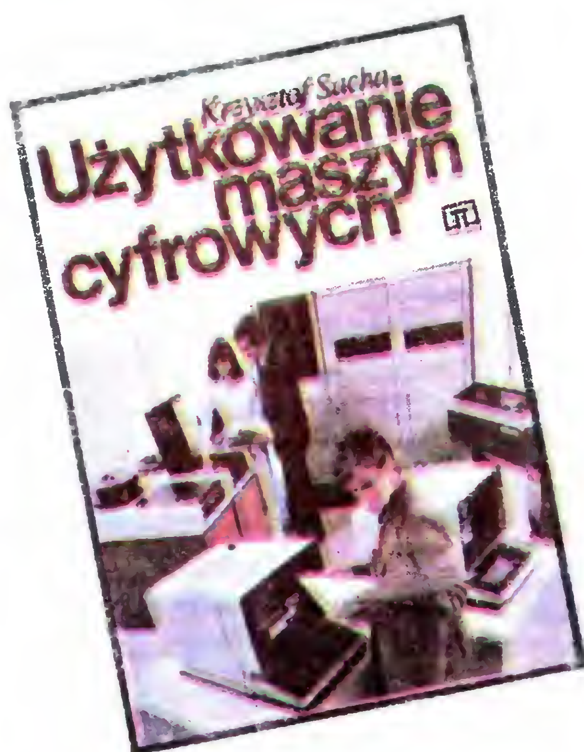
Mają Węgrzy również i swoje pisma mikrokomputerowe. Przedstawimy je w jednym z najbliższych numerów „Bajtka". Naszą szczególną sympatię wzbudził „Mikroszamitogep Magazin" (może dlatego, że tak pod względem graficznym jak i formalnym bardzo przypomina naszego „Bajtka").

Komputer można tu kupić w sklepie. W samy Budapeszcie jest co najmniej kilka, jeśli nie kilkanaście takich sklepów. Niektóre specjalizują się wyłącznie w komputerach i sprzęcie wideo, inne oferują te urządzenia obok np. aparatów fotograficznych telewizorów, gramofonów. Ceny są jednak bardzo wysokie. Oto kilka przykładów. ZX Spectrum 48 K — 17 tys. forintów, ZX Spectrum Plus — 25 tys., Commodore 64 — 30 tys., a C-128 — 50 tys., magnetofon do Commodore — 4 820 forintów, drukarka do tego komputera, MPS-803 — 34 tys. i wreszcie drążek sterowy Quick Shot II — 1 750 forintów.

Pomimo tak wysokich cen mikrokomputer coraz częściej trafia do domów węgierskich uczniów. Wszyscy jednak czekają na tani, krajowy komputer. Ale o tym w jednym z następnych „Bajtków".

Roman Poznański

WARTO PRZECZYTAĆ



UŻYTKOWANIE MASZYN CYFROWYCH

Przeznaczona dla IV i V klasy technikum elektronicznego książka **Krzysztofa Sachy „Użytkowanie maszyn cyfrowych"** po raz pierwszy wydana została w 1982 r. Poświęcona jest trzem głównym grupom zagadnień — programowaniu maszyn cyfrowych, ich zastosowaniu oraz obsłudze operatorskiej komputerów.

Niestety to, co młodzież obecnie najbardziej interesuje, czyli problematyka wykorzystania mikro- i minikomputerów poza ośrodkami obliczeniowymi, zostało w niej praktycznie całkowicie pominięte, nie licząc skromnego, półtorastronicowego dodatku na końcu książki. Sprzęt, na który autor powołuje się, a więc polskie komputery R-32 (głównie) oraz MERA-400 (dużo rzadziej) nie są bynajmniej ostatnią nowością nawet na krajowym rynku.

Natomiast cenną zaletą książki pana Sachy jest gruntowne uporządkowanie podstawowych pojęć. Na plus zapisać można również poświęcenie sporej partii tekstu programowaniu w języku FORTRAN.

Krzysztof Sacha „Użytkowanie maszyn cyfrowych", Warszawa 1985, Wydawnictwa Szkolne i Pedagogiczne, wyd. drugie.



MIKROKOMPUTER — PROGRAMOWANIE W JĘZYKU BASIC

„MIKROKOMPUTER — programowanie w języku BASIC" Bohdana Frelka i Andrzeja Lewandowskiego jest drugą z kolei pozycją wydawnictwa SIGMA z serii „micro-C".

O ile książka Wydawnictw Naukowo-Technicznych (recenzowana już w Bajtku) opisywała programowanie w języku BASIC w jego podstawowej wersji, o tyle ta pozycja wyraźnie ukierunkowana jest na programowanie Commodore 64, chociaż uwzględnione zostały również niuanse BASIC-a dla takich komputerów, jak: ZX 81, Spectrum, Meritum, VIC-20.

Zawiera ona dziewięć rozdziałów. Pierwszy, zatytułowany „Mikrokomputer — co to takiego?", wyjaśnia w przystępny sposób podstawowe pojęcia związane ze sprzętem i oprogramowaniem. Opisane zostały pokrótce takie pojęcia, jak: pamięć kasetowa, urządzenie dyskowe (disk drive), dyskietka, modem; pamięć RAM i ROM itp.

Rozdział drugi „Proste obliczenia" podpowiada, jak wykonać na komputerze operacje arytmetyczne. Autorzy zwracają np. uwagę na zagadnienia dokładności, precyzji liczb.

Obszerny, trzeci rozdział, to „Programowanie". Rozpoczyna się on od informatycznego abecadła, tj. od poznawania takich pojęć, jak linia programu, komentarz, zmienna prosta, słowa kluczowe (keywords), instrukcja podstawiania, zmienna sterująca, instrukcja skoku itd.

Kto opanował dobrze rozdział trzeci i ma w głowie jakiś ciekawy program, może teraz zająć się jego uruchomieniem. W tym celu zaglądamy do rozdziału czwartego: „Realizacja programu", gdzie uczymy się rozróżniać edytor liniowy od edytora ekranowego oraz jak unikać błędów syntaktycznych (składniowych) i błędów algorytmicznych (logicznych).

Rozdział piąty traktuje o operacjach na tekstach: jeśli chcemy założyć zbiór danych o utworach w domowej taśmotece, to na stronie 65 znajdujemy odpowiedni program, który — po odpowiedniej przeróbce — może być również wykorzystany do katalogowania książek, zbiorów filatelistycznych itp.

Rozdziały następne, tj. „Operacje wejścia-wyjścia", „Obraz i dźwięk" opisują dokładniej sposób korzystania z urządzeń peryferyjnych, budowanie grafiki niskorozdzielczej, poruszanie ruchomymi figurami (sprite'ami), sposób wysyłania sygnałów dźwiękowych. Przedostatni, ósmy rozdział „Wersje języka BASIC", opisuje podstawowe instrukcje omawianych wersji języka BASIC.

Pod koniec książki zamieszczono „Uwagi dla zaawansowanych", w których podano nieco informacji dotyczących budowy interpretera, a także sposoby optymalnego wykorzystania pamięci i konstruowania programów działających możliwie najszybciej.

Choć książka ta przeznaczona jest głównie dla posiadaczy C-64, z pewnością będzie pomocna również tym wszystkim, którzy stawiają dopiero swoje pierwsze kroki w programowaniu także na innych komputerach. Jest ona niezbędnym kompendium wiadomości dotyczących programowania.

„MIKROKOMPUTER — programowanie w języku BASIC": B. Frelek, A. Lewandowski, wyd. NOT — SIGMA; Warszawa 1986, wyd. I, nakład 60 370 egz.



WPROWADZENIE DO GRAFIKI KOMPUTEROWEJ

„Wprowadzenie do grafiki komputerowej" wydane w br. nakładem Wydawnictw Naukowo-Technicznych jest pierwszą, tak obszerną pracą, pozwalającą lepiej zrozumieć metody grafiki komputerowej. Po jej przeczytaniu i wykonaniu szeregu przykładów i ćwiczeń. Czytelnik może sam zaprogramować wykresy danych, wykresy kołowe, bryły obrotowe itp. Dzięki podanym wskazówkom możliwe jest też wykonanie bardziej skomplikowanych programów, np. programowe tworzenie rysunków technicznych.

Autor w sposób jasny i zrozumiały wprowadza Czytelnika w przekształcenia przestrzeni dwuwymiarowej, reprezentacje macierzowe, geometrię przestrzeni trójwymiarowej, obrazy perspektywiczne i stereoskopowe. Od Czytelnika wymaga się jedynie umiejętności posługiwania się językiem programowania Fortran IV i znajomości geometrii współrzędnych kartezjańskich.

Opisane programy można traktować jako zalążek pakietów graficznych, ale przede wszystkim są one środkami opisu algorytmów pozwalających rozwiązać dane zagadnienie. Można je łatwo przetłumaczyć na inne języki używane w grafice komputerowej, takie jak Pascal lub Basic. Podane procedury można również — niewielkim nakładem pracy — przystosować do dowolnych systemów graficznych: pisaków (ploterów) do papieru, monitorów ekranowych itp.

Głównym celem tej książki jest określenie podstaw grafiki komputerowej, przy jednoczesnym pozostawieniu jak największego pola dla wyobraźni użytkownika. Książka przeznaczona jest dla programistów, projektantów systemów przetwarzania informacji, pracowników nauki zajmujących się informatyką oraz dla studentów kierunków informatycznych.

Ian O.Angell — „Wprowadzenie do grafiki komputerowej", WNT, Warszawa 1986, s. 168, wydanie I, nakład 3800 egz., cena 140 zł

Jerzy Zawadzki

DWIE STRONY EKRANU

Jak nietrudno odgadnąć, chodzi oczywiście o ekran monitora podłączonego do mikrokomputera. Ekran, za którym zaczyna się świat pełen tajemnic, fascynujący urokiem nowości. Świat RAM-ów i ROM-ów, zamieszkały przez stworzenia, których imion próżno szukać w słownikach języka polskiego.

Czytając fachowe podręczniki lub słuchając rozmów prowadzonych przez tych którym udało się wedrzeć chociaż kawałek w głąb tajemniczej Komputerlandii trudno oprzeć się wrażeniu, że wszystko co siedzi tam w środku jest piekielnie skomplikowane. Trudno też nie dojść do wniosku, że moment, w którym komputer stanie się narzędziem pracy dostępnym dla każdego, jest jeszcze bardzo odległy. Może nawet wkład pracy potrzebny na sforsowanie szklanej przegrody oddzielającej świat ludzkich problemów od mówiącego tylko językiem dziwnych symboli świata drutów, jest zbyt duży. Może większość ludzi nie znajdzie dość czasu i zapału na to, by zapanowć nad komputerem i zmusić go do rozwiązywania swoich problemów.

Jest to czarna wizja i na szczęście zupełnie nierealna, z bardzo prostego powodu: większość typowych użytkowników korzysta z mikrokomputera znajdując się po tej drugiej (ludzkiej) stronie ekranu. Przeznaczone do rozwiązania zadania przedstawiają w języku zbliżonym do naturalnego języka ludzkiego i w takim samym języku otrzymują rozwiązania.

Bardzo często komputer dla nich to wypełniona jakąś tam elektroniką „czarna skrzynka" z klawiaturą, służącą do stawiania zadań, i ekranem, na którym pojawiają się wyniki. Ekranem, za którego nie zaglądają, bo nie muszą (o tym za chwilę) bo często nie mają na to czasu ani ochoty.

Myślę, że w tym miejscu niejeden z Czytelników — mikrofanów zapłonie oburzeniem: jak to, jak można nie mieć ochoty poznać działania urządzenia tak wspaniałego jakim jest mikrokomputer. Jak w ogóle można się posługiwać czymkolwiek nie zadając sobie trudu zrozumienia dlaczego to coś działa. Zanim jednak potępisz człowieka żyjącego przed ekranem popatrz dookoła. Czy jadąc samochodem przez most ludzie zastanawiają się jaka jest zasada działania silnika spalinowego, jak ropy naftowej otrzymać benzynę i co muszą zrobić inżynierowie-konstruktorzy żeby most wytrzymał ciężar pojazdu? Tak się składa, że każdy ma swoje hobby i każdy ma swoje narzędzia, od których wymaga tylko żeby były sprawne i niezawodne. I jeszcze tak się składa, że to co dla Ciebie jest fascynacją, dla większości ludzi jest (lub już wkrótce będzie) tylko narzędziem.

Od uświadomienia sobie tego faktu zależy bardzo wiele, przede wszystkim ułożenie poprawnych stosunków między Tobą, a zwykłymi użytkownikami sprzętu liczącego. „A dlaczego mamy utrzymywać poprawne stosunki? Niech się nauczą co to znaczy LXI B, OF1B i będziemy mieli o czym pogadać". Odpowiedź jest bardzo prosta: a dlatego, że jesteście sobie nawzajem potrzebni. Mianowicie, oni nie będą się uczyć co to znaczy adresowanie pośrednie, bo mogą skorzystać z Twojej wiedzy, natomiast Ty prędzej czy później dojdziesz do wniosku, że „sztuka dla sztuki" staje się nudna, że wszystko co wiesz powinno czemuś służyć. I wtedy zaczniecie ze sobą rozmawiać. O czym? O rzeczy najważniejszej dla użytkownika komputera czyli o programowaniu.

Przecież to właśnie oprogramowanie sprawia, że ludzie nie muszą znać całego wnętrza komputera po to by go z powodzeniem wykorzystywać. Ono akceptuje wystukiwane na klawiaturze liczby i teksty, tłumaczy je na język komputera, steruje procesem przetwarzania i ostatecznie zwraca wyniki znów przetworzone do „ludzkiej" postaci. A równocześnie PRODUKCJA oprogramowania jest ostatecznym celem, któremu służy coraz to lepsze poznawanie tajników komputera, a więc to, czemu poświęcasz teraz swój czas.

Po co o tym wszystkim piszę? Żeby uświadomić Ci, że mikrokomputer nie musi być tylko zabawką, że może stać się użytecznym narzędziem i to także dzięki Twojemu wkładowi. Po drugie, mam nadzieję, że dasz się namówić aby część swojego zainteresowania informatyką zostawić PRZED EKRANEM. Że oprócz myślenia jak coraz lepiej programować zechcesz czasem zastanowić się także nad tym CO programować i PO CO programować.

A skoro takie pytanie zostało już postawione, to warto wiedzieć, że już dziś życie udzieliło na nie wielu odpowiedzi. Istnieje już mnóstwo typowych zastosowań i typowego oprogramowania. Można nawet mówić o powstaniu pewnych nieoficjalnych standardów. Źle się stanie jeśli świat pójdzie tą drogą całkiem bez nas, dlatego „Bajtek" nie może pomijać milczeniem tak ważnej dziedziny „życia komputerowego".

Tym artykułem chcemy Cię zaprosić do rozmowy na temat wszystkiego, co — nazwijmy to umownie — dzieje się PRZED EKRANEM. Ze swej strony postaram się zaprezentować w „Bajtku" najbardziej typowe oprogramowanie oraz warunki, które we współczesnym świecie muszą spełniać programy, aby mogły stać się powszechnie używanym narzędziem. Mamy nadzieję, że zechcecie włączyć się do rozmowy i przedstawić własne propozycje wykorzystania mikrokomputera w Waszym codziennym, domowym, szkolnym czy klubowym życiu.

Andrzej Pilaszek

Ci z Was, którzy oglądali film pt. „Błękitny Grom" pamiętają być może scenę, w której lecący śmigłowcem bohaterowie uzyskują w ciągu kilkunastu sekund dane ściganego faceta, za pośrednictwem komputera pokładowego.

Niezależnie od wiarygodności całego filmu wspomniana scena nie ma w sobie nic z fantazji. Olbrzymie bazy danych, przechowujące miliony zapisów i dające dostęp do żądanych informacji w bardzo krótkim czasie stały się już powszechnym zjawiskiem. Bez użycia komputerów praktycznie nie potrafimy poradzić sobie z narastającą w oszałamiającym tempie lawiną informacji. Dlatego przechowywanie i przetwarzanie danych stało się jednym z głównych zastosowań maszyn cyfrowych, i to wcale nie tylko tych wielkich. Także dla mikrokomputerów pojawia się coraz więcej programów pozwalających budować i eksploatować bazy danych. Za chwilę opowiem o jednym z takich systemów, najpierw jednak kilka podstawowych informacji ogólnych.

Identyfikatory i rekordy

W ogromnym uproszczeniu bazy danych, które możemy spotkać na mikrokomputerach wyglądają następująco: podstawową porcją informacji jest tzw. REKORD (nazwa pochodzi od angielskiego „record", oznaczającego „zapis"). Rekordy zwykle zbudowane są z kilku — kilkudziesięciu pozycji czyli PÓL. W polu można zapisać jedną wielkość, np. liczbę, tekst, datę. Jedno (lub więcej) pól jest traktowane jako identyfikator rekordu — inaczej mówiąc, to co jest zapisane w tym polu jest wizytówką rekordu, czyli całej porcji informacji. Te wizytówki — identyfikatory pozwalają wybierać z bazy potrzebne porcje informacji — rekordy.

Trochę to skomplikowane, więc może posłużmy się prostym przykładem. Baza danych „SKLEROZA" przeznaczona jest do przechowywania danych o naszych przyjaciołach i znajomych. O każdym z nich chcemy pamiętać: nazwisko, imię, adres, numer telefonu, datę urodzenia.
Jeden rekord będzie zawierał pełną informację o jednym człowieku, musi więc składać się z pięciu pól i może wyglądać na przykład tak:

nazwisko	imię	adres	nr tel.	data ur.

A cała baza danych będzie to po prostu zbiór rekordów zapisanych w dowolnej (nie zanikającej po wyłączeniu!) pamięci komputera, np. na taśmie magnetycznej czy dyskietce. Niewielki fragment tej bazy może wyglądać na przykład tak:

nazwisko	imię	adres	nr tel.	data ur.
Kaczor	Donald	Hollywood	212121	01/01/57
Myszka	Miki	Pod Miotłą 5m.7	121212	31/12/55
Sowa	Kapitan	Na Tropie 8/9	333333	01/02/60

Identyfikatorami rekordów będą (niezależnie od siebie) pola: nazwisko i data ur. Oznacza to, że jeśli podamy nazwisko interesującego nas w danej chwili delikwenta, to po chwili na ekranie (ewentualnie na drukarce) powinny pojawić się wszystkie dane na jego temat. Analogicznie można wykorzystać drugi identyfikator: znając datę urodzenia osoby możemy natychmiast otrzymać pozostałe informacje o tej osobie.

Identyfikatory (często w literaturze zamiast słowa „identyfikator" używane jest słowo „klucz" — ang. key) służą do bardzo szybkiego wyszukiwania danych. Ale użycie klucza (identyfikatora rekordu) nie wyczerpuje wszystkich możliwości dostępu do danych. Można np. zażądać wyświetlenia wszystkich rekordów, które w polu „imię" mają wpisaną wartość „Andrzej", lub tych, w których nr tel. zaczyna się na 39. Wtedy dla uzyskania odpowiedzi przeglądane są wszystkie rekordy i wybierane te, które spełniają zadane warunki.

Na opisanie wszystkich możliwości dostępnych w typowej, działającej na mikrokomputerze, bazie danych nie starczy tu miejsca. Część z nich zobaczymy na przykładzie systemu SUPER-BASE 64.

Kto wykonuje wszystkie te funkcje? Oczywiście oprogramowanie. Stanowi ono, obok zbiorów rekordów, zasadniczą część systemu bazy danych. Tutaj uwaga na temat nazewnictwa: w zasadzie baza danych, to tylko fizyczny (zapisany na nośniku) zbiór rekordów. Gdy w grę wchodzi również oprogramowanie obsługujące ten zbiór powinniśmy mówić o „systemie bazy danych" (ang. database management system), jednak bardzo często mając na myśli cały system mówi się w skrócie baza danych.

Wracając do rzeczy, oprogramowanie zwalnia użytkownika od programowania transmisji danych między pamięcią operacyjną a urządzeniem, na którym są one przechowywane i bierze na siebie wszystkie kłopoty związane z właściwym rozmieszczaniem danych na nośniku.

Na zakończenie chciałbym zwrócić uwagę na bardzo ważną rzecz. Z systemem bazy danych (tak jak z każdym realizującym liczne funkcje programem użytkowym) musi być związany jakiś język. Przecież aby nasze życzenia mogły być zrealizowane muszą one najpierw zostać w jakiś sposób wyrażone. Oczywiście słowa „język" nie należy rozumieć całkiem dosłownie — chodzi o jakikolwiek system wymiany informacji, umownie nazywany językiem. Taki język musi umożliwiać zapisanie wszystkich poleceń, które umie wykonać oprogramowanie. Ponadto powinien być łatwy do opanowania i wygodny w użyciu — są to warunki ważne z punktu widzenia człowieka. Od ich spełnienia, czyli krótko mówiąc od jakości języka, zależy czy baza danych będzie dobrym, wygodnym narzędziem.

Inne, również ważne, cechy określające jakość systemów bazy danych to: szybkość działania, możliwość wymiany informacji z innymi programami, wreszcie pojemność, która w dobrych systemach jest już praktycznie limitowana tylko pojemnością użytych nośników magnetycznych.

A teraz, gdy już to wszystko wiemy, pora zapoznać się z jedną z lepszych spośród dostępnych na sprzęcie mikro baz danych, którą jest

SUPERBASE 64 dla COMMODORE C-64

Najpierw krótka metryczka: Superbase 64 jest systemem zarządzającym bazami danych na dyskach elastycznych, dla Commodore C-64. Minimalna potrzebna do uruchomienia systemu konfiguracja sprzętu, to mikrokomputer, monitor (telewizor) i pojedyńcza stacja dysków. Drukarka nie jest konieczna, ale przy poważniejszych zastosowaniach bardzo użyteczna.

W ubiegłym roku cena kompletu zawierającego dyskietkę z programem, podręcznik (ok. 250 stron!) i kasetę z nagranymi lekcjami korzystania z systemu, wynosiła a Anglii około 100 funtów.

W jednym zbiorze można zapisać aż do około 65000 rekordów. Zbiorów możemy mieć tyle, ile zmieści się na dyskietce. Oczywiście każdy z nich może zawierać rekordy o innej budowie. Zbiór jest zorganizowany tak sprytnie, że czas od momentu podania klucza do wyświetlenia zawartości rekordu nie przekracza 3 sekund — mimo że stacja dysków współpracująca z C-64 jest bardzo wolna.

Rekordy mogą być zbudowane z pól zawierających: teksty (do 255 znaków długości), liczby, daty i jeszcze parę innych trochę bardziej skomplikowanych rzeczy.

No właśnie, jak to jest z tymi polami? Budując naszą minibazę „SKLEROZA" określiliśmy od razu jak będą się nazywać pola i co będą zawierać. Ale przecież jeśli chcemy sprzedawać system bazy danych wielu ludziom, to nie jesteśmy w stanie z góry określić do czego będzie używana i jakie informacje przyjdzie jej przechowywać.

Rozwiązanie jest bardzo proste i typowe dla wszystkich sprzedawanych baz. System pozwala użytkownikowi zdefiniować taką strukturę rekordu, jaka jest potrzebna do konkretnego problemu. Po prostu, dla każdego pola mającego później wystąpić w rekordach zbioru danych trzeba najpierw podać jak się to pole będzie nazywać („nazwa pola"), jakie informacje mają być w nim zapisane („typ pola" np. tekstowe) i wreszcie jego długość, czyli ile maksymalnie znaków będzie można w tym polu zapamiętać.

W SUPERBASE budowę rekordu planuje się bardzo wygodnie, jeżdżąc kursorem po ekranie i naciskając kilka wyróżnionych klawiszy. W wyniku powstaje coś w rodzaju ramki określającej budowę rekordu. Taka pusta ramka będzie później wyświetlana gdy zechcemy zapisywać rekordy do zbioru. Wystarczy tylko w puste miejsca wpisać nasze dane i hop! już siedzą na dyskietce.

Przykład z BAJTKIEM

Jak teraz można z nich skorzystać? Znów posłużę się przykładem. Wyobraźmy sobie, że wpisaliśmy do bazy danych wszystkich prenumeratorów „Bajtka". Jest ich trochę mniej niż 65000 (nawiasem mówiąc szkoda, bo to świetny sposób na regularne otrzymywanie naszego pisma). Dla każdego z nich mamy zapisane między innymi wiek i płeć. Redaktor Naczelny pyta: jaka jest średnia wieku naszych prenumeratorów? Jedną (!) komendą możemy zliczyć równocześnie ilu jest ludzi i zsumować ich wiek. Wykonanie takiego polecenia trwa kilka do kilkunastu minut, w zależności od ilości wpisanych rekordów. Teraz pozostaje tylko podzielić drugą z otrzymanych liczb przez pierwszą.

Nie przedstawia to trudności, bo SUPERBASE współpracuje z interpreterem BASIC-u i potrafi wykonywać wszystkie obliczenia arytmetyczne (operacje tekstowe zresztą też). Następna sytuacja: chcemy poznać średnią wieku dziewcząt prenumerujących „Bajtka". Proszę bardzo, polecamy wykonać poprzednie obliczenie, ale tylko dla tych rekordów, w których w polu PŁEĆ jest wpisane „kobieta". Znowu kilkanaście minut i komputer wykonał pracę, która przeglądającej papierowe archiwum sekretarce zajęła by wiele godzin NUDNEJ i BEZMYŚLNEJ, a więc ciężkiej pracy. Poza tym, komputer na pewno nie pomylił się w sumowaniu!

Język SUPERBASE 64 zawiera około 40 komend. Jego ogromną dodatkową zaletą jest, wspomniana już, możliwość współpracy z BASIC-iem.

Znacznym udogodnieniem jest również tzw. MENU (znowu rzecz typowa, spotykana w wielu systemach). W oczekiwaniu na wydanie przez użytkownika nowej komendy system wyświetla spis najczęściej używanych, najpotrzebniejszych komend. Przy każdej z nich podana jest nazwa klawisza funkcyjnego. Nie trzeba sobie przypominać nazwy potrzebnej komendy ani tracić czasu na jej „wklepanie". Wystarczy popatrzeć na ekran i wcisnąć odpowiedni klawisz funkcyjny.

I jeszcze jedno typowe udogodnienie, które spotykamy również w SUPERBASE 64, czyli komenda HELP. Jeśli zapomniałeś szczegółów korzystania z komend nie musisz przerywać pracy i studiować podręcznika. Komendą HELP możesz poprosić o wyświetlenie najważniejszych informacji, potrzebnych do korzystania z systemu.

Wspomaganie przez program pamięci użytkownika nie jest jedynym przejawem życzliwego nastawienia systemu. Znów przykład: komenda LOAD „nazwa zbioru" przepisuje zbiór z dyskietki do pamięci operacyjnej. Jeśli w pośpiechu napiszemy LOAD i zapomnimy podać nazwę zbioru system nie odpisuje BLAD 129*33, bo tego nikt by nie zrozumiał bez czytania bardzo grubej książki. System uprzejmie prosi: „Podaj nazwę zbioru". Tego typu reakcji jest bardzo wiele. W ostatecznym rozrachunku ułatwiają i przyśpieszają wykonanie pracy i powodują, że jest nie tylko skuteczna, ale i przyjemna.

Nie święci garnki lepią

Oprogramowanie, nawet to najwspanialsze też tworzą zwykli ludzie. Może warto żebyś i Ty spróbował? Proponuję zaprojektowanie i oprogramowanie niewielkiej bazy danych, wspomagającej Twoją pamięć w wybranej dziedzinie życia codziennego. Może będzie to zbiór danych o wszystkich programach i podręcznikach posiadanych przez członków klubu, pozwalający szybko ustalić, czy ktoś ma potrzebny program i jak go można zdobyć, lub pomocny dla elektroników, zbiór danych o posiadanych częściach. Inna propozycja, to zbiór danych o wszystkich ludziach, z którymi utrzymujemy kontakty, pozwalający np. szybko stwierdzić kto ma dzisiaj urodziny.

A.P.



		BAJTKOWA GIEŁDA (tys. zł)	KOMIS (tys. zł)	BOMIS (tys. zł)	BALTONA PEWEX (Dol. USA)	RFN (średnie) (DM)	WLK. BRYT. (średnie) (£)
SINCLAIR	ZX 81	20–35	25–40	—	—	90	—
	ZX Spectrum 48 kB	70–80	105–110	130	125	150–250	45–65
	ZX Spectrum Plus	100–115	140–160	180	—	200–340	70–90
	ZX Spectrum 128 kB	195	—	—	—	520–590	150
	Drukarka SEIKOSHA GP50S	85–90	100	150	—	260	60–65
	Interface Kempston	8–16	20	—	—	40	7–9
	Magnetofon (średn. jak.)	20–35	40–45	40–60	—	50–120	15–20
COMMODORE	C–64	130–150	150–160	240	—	420–520	90–110
	C–128	280–350	450	450–500	—	700	210–230
	C–128 D	600	—	—	—	1500	390–410
	Magnetofon 1531	20–25	35	—	—	80–110	—
	Stacja dyskietek 1541	150–160	200	350	—	520	110–150
	Stacja dyskietek 1570	200	—	700	—	490–540	160–170
	Drukarka MPS 801	120–130	150	280	—	199	—
	Drukarka MPS 803	–	—	360	—	300–330	140–150
	Dyskietki 5 1/4 (średnia jak.)	0.7–1.8	1.2–1.5	2–3.5	—	1.2–5	0.8–2
ATARI	800 XL	70–80	90	—	115	140–180	70–80
	130 XE	140–160	190	380	199	360	150
	Stacja dyskietek 1050	130–140	200	—	185	478	130
	Magnetofon	25–30	35	—	48	75	20
	Drukarka 1029	140	200	—	199	—	85
AMSTRAD	464 z mon. monochromat.	190–220	300	500	—	680–720	160–180
	464 z monitorem kolor.	290	350	600	—	1100–1200	260–268
	664 z mon. monochromat.	300–350	—	600–700	—	1000–1300	—
	6128 z mon. monochromat.	370–400	700	850	—	1450–1500	250–260
	6128 z monit. kolor	485	—	1.1 mln.	—	1900–2000	340–350
	PCW 8256	900	1 mln.	1.2 mln.	—	2000–2100	380–400
	Dyskietki 3"	3.5–5	5–7	8	6	10–15	3.5–4

KLUB „MIKROS"

Klub Mikrokomputerowy „MIKROS" działa przy Zakładach Radiowych im. Marcina Kasprzaka w Warszawie. Członkami Klubu są pracownicy Zakładu oraz uczniowie szkoły przyzakładowej.

Pomysł założenia Klubu mikrokomuterowego w naszym zakładzie powstał pod koniec 1984 r. w niewielkim gronie pracowników Ośrodka Obliczeniowego URK. Potrzeba założenia klubu była ogromna, o czym świadczy fakt, że od powstania pomysłu do zebrania założycielskiego minął niespełna miesiąc. Na zebranie założycielskie przyszło 35 osób, po dwóch miesiącach działalności Klub liczył już około 150-ciu członków. W tym czasie nie dysponowaliśmy jeszcze żadnym komputerem, jedynie kilku naszych kolegów było szczęśliwymi posiadaczami ZX Spectrum. W związku z powyższym poszukiwaliśmy sponsora, który umożliwiłby nam zdobycie sprzętu. Pomimo istnienia w zakładzie kół stowarzyszeń technicznych i ekonomicznych, jedynie Zarząd Zakładowy ZSMP zgodził się objąć patronat nad naszym Klubem.

Pierwszym naszym posunięciem było zorganizowanie serii kursów programowania w języku BASIC na ZX SPECTRUM. Jak wszyscy wiemy, w tym czasie był to najbardziej popularny mikrokomputer domowy w Polsce. Przez pierwszy rok naszej działalności, mieliśmy dostęp jedynie do tego typu sprzętu. Od czasu upowszechnienia się ATARI w Polsce, wzrosło nasze zainteresowanie komputerami tej marki. Nawiązaliśmy współpracę z firmą PZ KAREN, która udostępniła nam ATARI 800XL. Mamy nadzieję, że w przyszłości będziemy mogli korzystać również z komputerów serii ST. W zdobywaniu oprogramowania i dokumentacji w języku polskim pomaga nam także firma EUROBIT z Warszawy.

Od września 1985 r. rozpoczęliśmy współpracę z Telewizją Polską poprzez udział członków naszego Klubu w programie „Majsterklepka". Prezentujemy w nim pisane przez nas programy, dotyczące zastosowań mikrokomputerów w nauce i technice. Oprócz demonstracji działania programów, oma-

wiane są także ich algorytmy oraz możliwości modyfikacji. W czasie trwania audycji, programy te emitowane są na falach UKF. Pozwala to telewidzom z całej Polski na nagranie ich i wykorzystywanie na własnych komputerach.

Po rozpoczęciu emisji programów komputerowych napłynęło do nas bardzo dużo listów. Proszono w nich o pomoc w nawiązaniu kontaktów z innymi użytkownikami mikrokomputerów, zakładaniu własnych klubów, wymianie oprogramowania i literatury. W celu zintegrowania fanów mikrokomputerów powołaliśmy za pośrednictwem Telewizji „Korespondencyjny Klub Komputerowy". Ilość zgłoszeń do nowo powołanego klubu przerosła nasze oczekiwania. Przychodziły do nas listy z najdalszych zakątków Polski, a ostatnio także z Czechosłowacji. Na wszystkie z nich staramy się sukcesywnie odpowiadać.

Jako zawodowi informatycy wykorzystujący w pracy zawodowej mikrokomputery typu IBM PC, służymy fachową radą i pomocą pozostałym członkom Klubu. Naszą naczelną intencją jest uświadomienie kadrze inżynierskiej zakładu efektów płynących z zastosowania mikrokomputerów w miejscu pracy. Poprzez demonstracje sposobów wykorzystania mikrokomputerów domowych uświadamiamy im możliwości jakimi dysponują mikrokomputery profesjonalne. W przyszłości chcielibyśmy demonstracje te prowadzić również na sprzęcie profesjonalnym. Członkowie naszego Klubu reprezentują prawie wszystkie wydziały zakładu. Stąd mamy nadzieję, że nowatorska myśl informatyczna znajdzie zastosowanie w całym przedsiębiorstwie.

W swych działaniach staramy się nie zapominać także o najmłodszych. Organizujemy dla nich „Ferie z komputerem", umilamy im pobyt na wakacjach w zakładowym ośrodku wczasowym. Poprzez zabawę staramy się przybliżyć im informatykę. Prowadzimy prelekcje i pokazy w domach kultury, szkołach a nawet w przedszkolach.

Po prawie półtorarocznej działalności Klubu, dysponujemy obecnie: dwoma komputerami MERITUM I oraz komputerem UNIPOLBRIT 2086 wraz z drukarkami SEIKOSHA GP500 AS zakupionymi przez nasz Zakład, oraz ATARI 800XL ze stacją dyskietek, magnetofonem i drukarką udostępnionymi nam przez firmę PZ KAREN z Warszawy. Posiadamy również bogatą bibliotekę oprogramowania dla komputerów ZX SPECTRUM, MERITUM oraz ATARI.

Chcielibyśmy podkreślić, że od samego początku naszego istnienia spotkaliśmy się z dużą pomocą ze strony dyrekcji Zakładu, co pozwoliło nam przezwyciężyć wiele trudności.

Działalność nasza nie jest pozbawiona jednak problemów. Najważniejszym z nich jest po prostu brak czasu. Zajęcia klubowe prowadzone są przez nas w czasie wolnym od pracy, a tego mamy coraz mniej.

ADRES DO KORESPONDENCJI:

Klub Mikrokomputerowy „MIKROS"
Zakłady Radiowe im. M. Kasprzaka
01-211 Warszawa

Albert Kłóskiewicz
Piotr Pieczerak

KLAN NIETYPOWYCH

ORIC-1

Jesteśmy wraz z kolegą posiadaczami i użytkownikami mikrokomputera domowego ORIC-1 z pamięcią 64 K. Dysponujemy szeregiem ciekawych programów na ten komputer, w tym także monitorem i asemblerem o nazwie ORICMON. Stworzyliśmy również kilka gier i programów użytkowych jak np. ORIC BRIDGE i ORIC LOGO.

Obecnie poszukujemy kontaktu z innymi użytkownikami komputera ORIC-1 celem wymiany programów i literatury. Oprócz podanego adresu możliwy jest również kontakt telefoniczny: Gdynia, 21-70-41 w. 139 w godz. 9 do 14-tej.

Stefan Kluj
ul. Gdańska 6 m 65
84-230 RUMIA

SPRZĘŻENIE ZWROTNE



Drogi Bajtku!

Czy pomiędzy komputerami Commodore 16 i 64 zachodzi podobna relacja, co między Spectrum 16 i 48? To znaczy — czy są to typy kompatybilne?

Marian Herbst
ul. Grochowska 98 m. 14
Warszawa

Nie. Pomimo zewnętrznego podobieństwa, są to zupełnie inne komputery. Zgodność zachodzi jedynie pomiędzy C16 i C116 (ten ostatni ma po prostu tańszą, za to znacznie gorszą klawiaturę), oraz — w jedną stronę — pomiędzy C 64 i C 128. Podobnie sprawa ma się z Commodore +4, który z kolei jest niezgodny z żadnym z wymienionych typów.

Poprzez znajomego mam możliwość zdobywania doświadczeń programistycznych na dużym komputerze Odra 1300. W grę wchodzi język programowania FORTRAN. Prosiłbym o informację n/t dostępnej literatury.

Krzysztof Czyż
(adres do wiadomości redakcji)

Oprócz skryptów różnych uczelni i instytucji, dotyczących programowania w Fortranie, znane mi są dwie książki, zawierające opis tego języka:

— J. Biańkowski, K. Fijałkowski, Z. Odrowąż–Sypniewski „Programowanie w języku Fortran", wyd. V, PWN 1978;

oraz

— W. Pachelski „Fortran IV dla maszyn Jednolitego Systemu", wyd. I, WNT 1981.

Bardziej godną polecenia wydaje się pozycja druga. Nie polecam natomiast samego Fortranu; obecnie jest to już prehistoria informatyki, język trudny i niewygodny w użyciu, choć nadal używany w wielu ośrodkach obliczeniowych, zwłaszcza na sprzęcie z ubiegłego dziesięciolecia. Być może warto dołożyć starań, by doświadczenia zdobywać za pomocą innego narzędzia?

Posiadam od kilku tygodni komputer Amstrad CPC 464 oraz drukarkę. Ta ostatnia jest pożytecznym uzupełnieniem samego komputera, dodatkowo potrzebnym mi w pracy zawodowej. Jednak mam pewien kłopot: listing programu „wyrzucony" na drukarkę ma za każdym razem po 40 znaków w linii, niezależnie od tego, czy na ekranie mam tych znaków 20, 40 czy 80. Pozostaje w ten sposób z prawej strony margines szerokości ok. 1/3 strony. Marnotrawstwo papieru! Czy można to zmienić?

S.L.
(nazwisko i adres do wiadomości redakcji)

Sprawa jest prosta. Basic Amstrada zawiera instrukcję

WIDTH n

która określa szerokość wiersza drukarki na n znaków. W niektórych programach użytkowych szerokość wiersza jest definiowana automatycznie (np. Dr Logo), i wtedy uprzednie ustalenie szerokości w powyższy sposób nie wystarczy.

Chciałbym przedstawić sposób dzielenia obrzeża ekranu Spectrum na części. Program jest cały w Basicu.

Przykład 1

```
1 BORDER 1: LET a=1: BORDER 6: GO TO 1
```

Przykład 2

```
2 BORDER 1: PAUSE 1: BORDER 6:
LET a=1: LET a=1 LET a=1 GO TO 2
```

Robert Wańkowski
ul. Spokojna 3a
62–020 Swarzędz

Działanie powyższych programików opiera się na pewnym dowcipnym pomyśle, dlatego [illegible] jąc jednocześnie do czytelników pytanie: dlaczego faktycznie dzielą one obrzeże ekranu na części?

Czym różnią się instrukcje TS i CT w Sinclair Logo? Osobiście nie zauważyłem żadnej różnicy.

Rafał Jesionowski
Kraków

TEXTSCREEN (w skrócie TS) powoduje inicjalizację ekranu tekstowego, natomiast CT — usuwa tekst z ekranu. W trybie tekstowym faktycznie działanie obu tych procedur jest identyczne. Aby zauważyć różnicę, należy wykonać:

CS
TS

i dla porównania:

CS
CT

W drugim przypadku usunięty zostaje tekst z dolnych linii ekranu, bez niszczenia rysunku.

Faktem jest natomiast, że przy dwóch liniach tekstu w trybie graficznym Sinclair LOGO różnica jest raczej symboliczna.

Podobno można w jakiś sposób zrealizować grafikę wysokiej rozdzielczości (256 × 196 punktów) na ZX 81 (16 kB RAM) bez przystawki HRG. Jak?

Dariusz Schindzielorz
ul. Akacjowa 6a m. 7
47–330 Zdzieszowice

Istnieją dwa sposoby uzyskania grafiki wysokiej rozdzielczości na ZX 81: programowy lub sprzętowy. Przykładem pierwszego rozwiązania jest program o nazwie HRG, pozwalający na uzyskanie wymienionej rozdzielczości, wszakże z wyłączeniem niektórych punktów. W rozwiązaniu sprzętowym można natomiast obejść się bez dodatkowych urządzeń. Możliwa jest bowiem taka zmiana połączeń wewnętrznych, która pozwala na przechowywanie wzorców znaków w pamięci RAM. Pozwala to na dowolne ich deklarowanie, a co za tym idzie — pozwala uzyskać efekty graficzne. Rozwiązanie takie zostało opisane w dwóch numerach mikroKLANu: 14 i 19/1985, stanowiących wkładki do miesięcznika „Informatyka" nr 2/85 i 7/85. Powyższe możliwości dotyczą tylko ZX 81 16 kB.

Ułożyłem kilka gier zręcznościowych na ZX Spectrum. Zawsze sterowanie bohatera odbywało się przez klawiaturę (z zastosowaniem funkcji INKEY$). Chciałbym, aby sterowanie odbywało się także za pomocą joysticka typu Kempston. W jaki sposób mogę to zrealizować?

Ireneusz Tabor
ul. Lisowiecka 18/1
42–700 Lubliniec

W przypadku interface'u typu Kempston należy funkcję INKEY$ zastąpić funkcją IN 31. Bardzo łatwo jest dowiedzieć się, jakie wartości tej funkcji odpowiadają różnym kierunkom wychylenia drążka. Umożliwia to poniższy program:

```
10 REM Joystick Kempston
20 REM --- TEST ---
30 PRINT AT 2,2;IN 31;"  "
40 GO TO 20
```

Marcin Waligórski

WSZYSTKO DLA WSZYSTKICH

ZANIM KUPISZ KOMPUTER ZADZWOŃ 28-01-76

KUPIŁEŚ KOMPUTER ZADZWOŃ 28-01-76

eurobit

PORADY:
— WYBÓR SPRZĘTU
— ZASTOSOWANIA
OPROGRAMOWANIE
INSTRUKCJE
POŚREDNICTWO
ZLECENIA

ATARI
COMMODORE C-16, 116, 4
AMSTRAD
VIC 20; C-64
SHARP
MSX
IBM

WYSTAWIAMY RACHUNKI DLA INSTYTUCJI
Al. Ujazdowskie 18 m. 14 Warszawa w g. 9-18

D-90

DŻOJSTIKI — „SUGUZ"

DOMY TOWAROWE „CENTRUM"
„JUNIOR" — STOISKO RTV

Adres producenta:
05-805 Kanie,
ul. Kolejowa 12b
(TEL. DOM. 46-39-48).

D-95

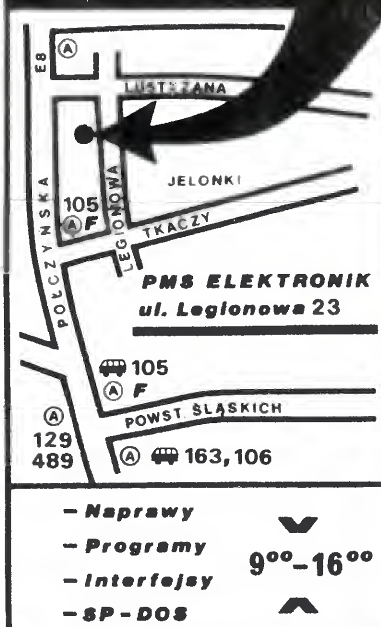
ZELMEVAC
service
COMPUTECHNIK

SERWIS GWARANCYJNY, INFORMACJE I DEMONSTRACJE KOMPUTERÓW:

— Commodore C-64
— Commodore C-128
— Schneider CPC 464
— Schneider CPC 6128
— Joyce PCW 8256 i PCW 8512
— Urządzenia peryferyjne
— Dyskietki, kasety,
— Gry, programy

01-793 Warszawa, ul. Rydygiera 9c
Tel.: 39-05-64 w godz. 9-15 oprócz sobót.

D-73



JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy), 04-028 Warszawa, Al. Stanów Zjednoczonych 53, pokój 313.
Tel. 10-56-82.

Cena ogłoszeń: 200 zł za 1 cm plus dodatki za kolor.

SPECJALNA OFERTA DLA: INSTYTUCJI, SZKÓŁ, OŚRODKÓW KULTURY, KLUBÓW, OSÓB PRYWATNYCH.

ENTER computing — wypożycza za zaliczeniem pocztowym wszechstronne oprogramowanie ZX SPECTRUM
02-105 W-wa 21 P-3
Termin realizacji zamówienia: 5 dni.
Wyczerpujące informacje po nadesłaniu koperty zwrotnej.

D-101

Instalacje systemów przyśpieszających współpracę ze stacją dysków do COMMODORE — 64, CROS-ASSEMBLER Z — 80 NA COMMODORE 64
Informacje korespondencyjnie: 50-012 Wrocław, ul. Kościuszki 40/3.

D-96

ATARI. INTERFEJS DO ZWYKŁEGO MAGNETOFONU ZASTĄPI MAGNETOFON FABRYCZNY.
INFORMACJE:
WARSZAWA 34-16-06.

D-83

Programowanie pamięci EROM, PROM. Naprawa i rozbudowa mikrokomputerów. Wykorzystanie popularnych mikrokomputerów w laboratoryjnych systemach pomiarowych i układach kontrolno-sterujących. Rakowski Marian, Bydgoszcz, ul. Żmudzka 68b, tel. 42-39-05.

D-100

9 W NASTĘPNYM NUMERZE

- Doc. Stanisław Waligórski przedstawia program „Elementy informatyki".
- TEST drukarki „Gemini 10X"
- Apple II na Grzybowskiej
- Jak pisać dobre programy
- Co jest grane? PANAMA JOE!
- Laser 128
- NOTESIK dla przedszkolaków
- KLANY: Commodore, Amstrad — Schneider, Atari, Spectrum
- Jednoręki bandyta w Twoim domu!

ABY ZAPRENUMEROWAĆ BAJTKA MUSISZ WIEDZIEĆ!

- **jakie są terminy dokonywania wpłat**
  - do dnia 10 listopada: na styczeń, I kwartał, I półrocze roku następnego oraz cały rok następny
  - do dnia 1-go tego miesiąca, który poprzedza okres prenumeraty (roku bieżącego)
- **jakie są ceny**
  - kwartał: 300 zł, półrocze: 600 zł, rok: 1200 zł
- **jak mogą dokonać prenumeraty osoby prywatne**
  - czytelnicy mieszkający w miastach, gdzie istnieją Oddziały RSW „Prasa-Książka-Ruch" opłacają prenumeratę w urzędach pocztowych nadawczo-oddawczych (właściwych dla miejsca zamieszkania prenumeratora) wpłacając na rachunek bankowy miejscowego Oddziału RSW „Prasa-Książka-Ruch"
  - czytelnicy mieszkający na wsi i miejscowościach, gdzie nie ma Oddziałów RSW „Prasa-Książka-Ruch" opłacają prenumeratę w urzędach pocztowych i u doręczycieli
- **jak mogą dokonać prenumeraty instytucje i zakłady pracy**
  - zamówienia na prenumeratę przyjmowane są bezpośrednio przez Oddziały RSW „Prasa-Książka-Ruch" — a w miejscowościach, gdzie nie ma Oddziałów RSW prenumeraty można dokonać w urzędach pocztowych
- **jak otrzymać BAJTKA za granicą**
  - prenumeratę ze zleceniem wysyłki przyjmuje RSW „Prasa-Książka-Ruch" Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto NBP XV Oddział w Warszawie Nr 1153-201045-139-11. Prenumerata ta, (z wysyłką pocztą zwykłą) jest droższa od prenumeraty krajowej o 50% dla osób prywatnych i o 100% dla instytucji i zakładów pracy

Ten oto blankiet przekazowy pomoże Ci zaprenumerować BAJTKA. A więc wpisz czytelnie swoje imię, nazwisko i adres, starannie wytnij przekaz, udaj się z nim na pocztę, tam uzupełnij numer rachunku i siedzibę Twojego Oddziału RSW „Prasa-Książka-Ruch" i... BAJTEK sam trafi do Twojego domu!!!

POTWIERDZENIE DLA WPŁACAJĄCEGO
SŁOWNIE ZŁOTYCH
ZŁ
NAZWISKO I IMIĘ
ADRES
ULICA, NR DOMU I MIESZKANIA
KOD POCZTOWY
MIEJSCOWOŚĆ
NAZWA I SIEDZIBA POSIADACZA RACHUNKU
DO WPŁATY NA R-K NR
W NBP ODDZIAŁ W
PODPIS PRZYJ.
OPŁATA
ZŁ
DATOWNIK
PRENUMERATA PRASY

ODCINEK DLA POSIADACZA RACHUNKU
NR ZAMÓWIENIA
SŁOWNIE ZŁOTYCH
ZŁ
NR ODDZIAŁU RSW
NR ODBIORCY
NAZWISKO I IMIĘ
ADRES
ULICA, NR DOMU I MIESZKANIA
KOD POCZTOWY
MIEJSCOWOŚĆ
NAZWA I SIEDZIBA POSIADACZA RACHUNKU
DO WPŁATY NA R-K NR
W NBP ODDZIAŁ W
STEMPEL OKRĘGOWY
UPT NADAWCZO-ODDAWCZEGO
DATOWNIK
PRENUMERATA PRASY

ODCINEK DLA POCZTY
SŁOWNIE ZŁOTYCH
ZŁ
NAZWISKO I IMIĘ
ADRES
ULICA, NR DOMU I MIESZKANIA
KOD POCZTOWY
MIEJSCOWOŚĆ
NAZWA I SIEDZIBA POSIADACZA RACHUNKU
DO WPŁATY NA R-K NR
W NBP ODDZIAŁ W
PODPIS PRZYJ.
OPŁATA
ZŁ
DATOWNIK
PRENUMERATA PRASY
PRZEKAZ DLA WPŁAT NA RACHUNKI W NBP

RUBRYKI ZAZNACZONE GRUBĄ LINIĄ NA DWÓCH STRONACH BLANKIETU WYPEŁNIA WPŁACAJĄCY

Bajtek | TYTUŁ | ILOŚĆ PREN | OKRES PRENUMERATY OD M-CA DO M-CA | WARTOŚĆ | RAZEM

MIEJSCE ODBIORU

NR INDEKSU 353965

Bajtek | TYTUŁ | ILOŚĆ PREN | OKRES PRENUMERATY OD M-CA DO M-CA | WARTOŚĆ | RAZEM

BLANKIET PRZEKAZOWY PROSIMY WYPEŁNIĆ DOKŁADNIE I CZYTELNIE

Bajtek | TYTUŁ | ILOŚĆ PREN | OKRES PRENUMERATY OD M-CA DO M-CA | WARTOŚĆ | RAZEM

TAJEMNICZY ZAMEK

Cześć Maluchy!

Widzieliście grę pod tytułem Hobbit? Jeśli nie, to możecie przeczytać o niej w drugim numerze „Bajtka" z tego roku. Takie gry nazywa się najczęściej angielskim słowem „adventure" czyli przygodowe.

Zasada jest tu zupełnie inna niż w grach z rodzaju „bij zabij" czy „łap i uciekaj". Małpia zręczność i refleks nie są potrzebne. Należy za to trochę ruszyć głową. Najczęściej gry przygodowe nie posiadają prawie wcale grafiki. Na ekranie ukazują się opisy kolejnych sytuacji. Do ciebie należy znalezienie z nich wyjścia. Sukces polega na odkryciu toku postępowania, który prowadzi do wypełnienia postawionego przed tobą zadania.

Wydaje się, że tego rodzaju programy muszą być bardzo złożone. Niektóre z nich rzeczywiście zajmują po kilkadziesiąt kilobajtów. Można jednak dla zabawy spróbować napisać własną, prostą grę przygodową.

Oczywiście jak zawsze najważniejszy jest pomysł. Resztę można zrobić za pomocą czterech instrukcji — PRINT, INPUT, IF...THEN... i GOTO. Nie wierzycie? Popatrzcie więc na wydruk programu. No tak, rzeczywiście, jest jeszcze jedna instrukcja. W linii 650 skorzystaliśmy z OR czyli „lub". Oszczędza nam to kilku dodatkowych linii programu.

Przy okazji zwróćcie uwagę na tekst drukowany na ekranie. Został on tak napisany, że nie ma w nim liter ą, ę, niedostępnych bezpośrednio z klawiatury komputera. Wymyślenie takiego tekstu może być samo w sobie niezłą zabawą, choć z programowaniem nie ma nic wspólnego. Ci z was, którzy potrafią zdefiniować własne litery na swoim komputerze nie potrzebują oczywiście korzystać z tej metody. (Sposoby definiowania własnych znaków dla poszczególnych mikrokomputerów prezentowaliśmy już w „Bajtku" kilkakrotnie i będziemy wracać jeszcze do tego tematu.)

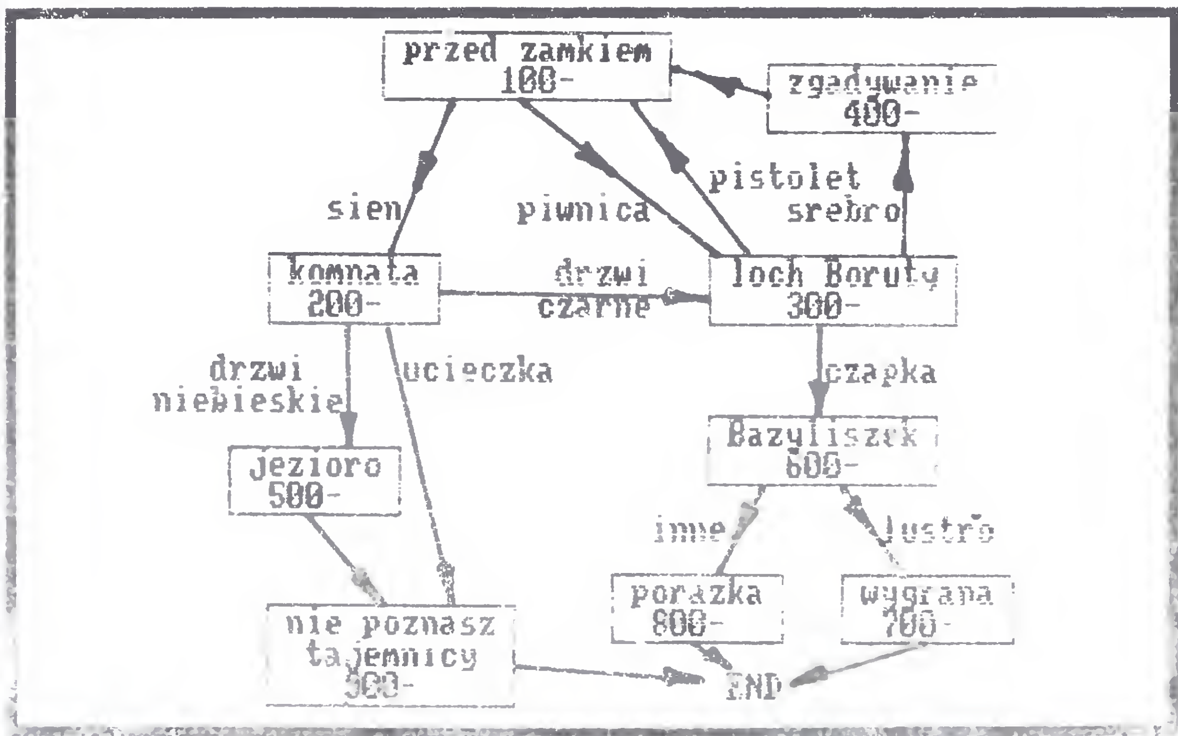
Teraz wróćmy jednak do naszego programu. Popatrzmy jak działa. Składa się on z kilku podobnych części. Najpierw drukowany jest opis sytuacji (wraz z podanymi możliwościami wyboru) poniżej, przy pomocy instrukcji INPUT, odczytywana jest odpowiedź, której udzielił grający. W następnej kolejności realizowane są dwie (lub więcej) instrukcje warunkowe, które powodują skok (GOTO) do odpowiedniej części programu w zależności od tego, co wybrał grający.

Warto — przy okazji — zawrzeć bliższą znajomość z nową instrukcją: OR. Słowo to oznacza w języku angielskim „lub" i dokładnie tak możemy ją odczytywać analizując programy. Powiedzmy, że piszemy program, w którym obliczamy wartość zmiennej A. Jeśli na koniec A będzie równe 5 albo też 10 powinien ukazać się komunikat "DOBRZE". Możemy to zrobić tak:

1000 IF A=5 THEN PRINT "DOBRZE"

1010 IF A=10 THEN PRINT "DOBRZE" Wygodniej jednak będzie użyć jednego zdania:

1000 IF A=5 OR A=10 THEN PRINT "DOBRZE" Czyli: jeśli A jest równe 5, lub A jest równe 10 wtedy pisz "Dobrze".



DLA PRZEDSZKOLAKÓW



I jeszcze jedna podobna instrukcja AND czyli "i". Stosujemy ją wówczas, gdy chcemy np. by komunikat "DOBRZE" ukazał się w przypadku gdy A=5 i równocześnie B=10. Bez instrukcji AND wygląda to tak:

1000 IF A=5 THEN IF=10 THEN PRINT "DOBRZE"

Natomiast z AND:

1000 IF A=5 AND B=10 THEN PRINT „DOBRZE"

Dysponując taką wiedzą możemy już zabrać się do pisania programu. Może on zawierać nie kilka (jak nasz przykład) części, a kilkanaście, kilkadziesiąt, a jeśliby ktoś miał wystarczająco dużo cierpliwości to nawet kilkaset. Najpierw należy jednak wymyśleć odpowiednią fabułę i narysować schemat blokowy — podobny do tego, który przedstawiony jest na rysunku — i oznaczyć „przejścia". W tym momencie napisanie samego programu nie jest już żadnym problemem. Po raz kolejny potwierdzi się prawda, że w naszym programie najważniejszy jest pomysł, reszta to tylko rzemiosło.

Spróbujcie wykorzystać ten pomysł do napisania własnej gry. Przyślijcie ją do „Bajtka". Pamiętajcie, że poczucie humoru cenne jest na równi z umiejętnością programowania. Najbardziej dowcipnych czekają nagrody.

Romek

```
10 PRINT "TAJEMNICZY ZAMEK"
20 PRINT
30 PRINT "Za siedmioma morzami,
za siedmioma lasami stoi ponure
zamczysko. Kto tylko wejdzie w j
ego mury ginie bez ratunku. Ryce
rze z wielu krain przybywali pod
 bramy warowni lecz tajemnicy ni
e odkryli."
40 PRINT "Teraz twoja kolej!"
100 PRINT
110 PRINT "Stoisz pod murami zam
ku. Widzisz drzwi do piwnic (p)
i schody do sieni zamkowej (s).
Wybieraj!"
120 INPUT a$
130 IF a$="p" THEN GOTO 300
140 IF a$="s" THEN GOTO 200
150 GOTO 100
200 PRINT
210 PRINT "Wielka, jasna komnata
. Dwoje drzwi: niebieskie (n) i
czarne (c).Tajemniczy szept radz
i ci uciec (u). Wybieraj!"
220 INPUT a$

210 PRINT "Wielka, jasna komnata
. Dwoje drzwi: niebieskie (n) i
czarne (c).Tajemniczy szept radz
i ci uciec (u). Wybieraj!"
220 INPUT a$
230 IF a$="n" THEN GOTO 510
240 IF a$="c" THEN GOTO 300
250 IF a$="u" THEN GOTO 900
260 GOTO 200
300 PRINT
310 PRINT "Ciemny, wilgotny loch
. W rogu, pod drzwiami okrutnie
chrapie czart Boruta. Obok  stoi
 worek srebra (w), na nim pistol
et (p) i czapka (c). Zabierz prz
edmiot najcenniejszy dla czarta.
 W zamian za ten przedmiot wejdz
iesz do kolejnej komnaty."
320 INPUT a$
330 IF a$="w" THEN GOTO 410
340 IF a$="p" THEN GOTO 100
350 IF a$="c" THEN GOTO 600
360 GOTO 300
400 PRINT
410 PRINT "Wyrwany ze snu Boruta
 zaciska pazury na twoim karku.
Przekroczysz progi kolejnej komn
aty gdy odpowiesz na pytanie: il
e sztuk srebra jest w worku. Zga
duj! (liczba)"
420 INPUT a
430 PRINT "Nieprawda - krzyczy B
oruta. Uciekasz.
440 GOTO 100
500 PRINT
510 PRINT "Za niebieskimi drzwia
mi ukryte jest zaczarowane jezio
ro. Uwalniasz jego wody i zamek
ginie pod wzburzonymi falami."
520 GOTO 900
600 PRINT
610 PRINT "Boruta nigdy nie opus
zcza lochu bez czapki. Ukrywa on
a przed ludzkim wzrokiem jego di
abelskie rogi. Za zwrot czapki B
oruta otwiera ci drzwi do kolejn
ej komnaty."
620 PRINT

630 PRINT "Szeroki korytarz. Poc
hodnie (p) przytwierdzone do mur
u. Obok szabla (s), maczuga (m)
i kusza (k). Z drugiej strony lu
stro (l) i zbroja rycerska (z).
Nadchodzi straszny Bazyliszek. W
ybierz jeden przedmiot i zabij p
otwora."
640 INPUT a$
650 IF a$="p"OR a$="s" OR a$="m"
 OR a$="k" OR a$="z" THEN GOTO 3
30
660 IF a$="l" THEN GOTO 700
670 GOTO 600
700 PRINT
710 PRINT "Bazyliszek pada martw
y na widok swego odbicia w lustr
ze. Boruta umyka gdzie pieprz wy
rasta. Odwieczny czar znika. Nad
chodzi prawdziwa pani zamku, dzi
ewczyna cudnej urody. I jak to w
e wszystkich bajkach bywa - jutr
o wasze wesele."
720 PRINT "Tym razem nie masz wy
boru!"

730 GOTO 1000
800 PRINT "Nic ci po takiej bron
i. Bazyliszek zabija swe ofiary
wzrokiem"
900 PRINT "Tajemnica zamku jest
nadal niepoznana."
1000 END
```

NIE TYLKO KOMPUTERY

dokończenie ze str. 32

nomia optyczna, również radioastronomia stanęła w swym rozwoju przed murem nie do przebicia. Mur ten jednak przekroczono — dzięki geniuszowi ludzkiemu, matematyce i komputerom. Jeśli ktoś uważał do tej pory matematykę tylko i wyłącznie na nudy i nikomu niepotrzebny przedmiot szkolny — to ma właśnie okazję zmienić zdanie!

Okazuje się, że zamiast budować coraz większe anteny, wystarczy odpowiednio skorelować obserwacje prowadzone przez anteny mniejsze. Następnie, za pomocą niezbyt skomplikowanych obliczeń (dwuwymiarowe odwzorowanie Fouriera) otrzymuje się obraz równoważny temu, jaki otrzymalibyśmy dysponując anteną o średnicy równej odległości pomiędzy antenami mniejszymi. Jest to zupełnie fantastyczna możliwość, gdyż oznacza przecież, że używając do obserwacji dwóch anten odległych od siebie na przykład o 100 km, dysponujemy faktycznie zupełnie niemożliwością dzisiaj do zbudowania anteną, której czasza miałaby 100 km średnicy!

Liczbę powyższą można dowolnie zwiększać. I to się robi. We wspólnym eksperymencie przeprowadzonym przez obserwatoria w Green Bank (USA) i w Simeiżk (ZSRR, Krym) uzyskano np. pozytywne wyniki na bazie 8030 km.

Za opracowanie i zrealizowanie tej właśnie metody obserwacji — nazwanej supersyntezą aparatury — prof. Martin Ryle otrzymał w 1978 roku jak najbardziej zasłużoną Nagrodę Nobla.

A wyobraźmy sobie, że z radioteleskopem na Ziemi współpracował będzie radioteleskop umieszczony w przestrzeni kosmicznej...

RADIOSTRON

Idźmy dalej. A co stoi na przeszkodzie aby stworzyć przyrząd (nazywa się on fachowo interferometrem) złożony z dwóch lub więcej radioteleskopów umieszczonych w kosmosie? Przecież będziemy mogli dysponować dzięki temu takimi możliwościami, jakie uzyskalibyśmy budując na przykład antenę paraboliczną o średnicy ponad 70 tysięcy kilometrów (średnica orbity geostacjonarnej). Jest to już dzisiaj w pełni realne.

Akademik Roald Sagdiejew, dyrektor radzieckiego Instytutu Badań Kosmicznych zapowiedział, że taki właśnie interferometr, złożony z trzech umieszczonych w kosmosie radioteleskopów, Związek Radziecki zamierza uruchomić w roku 1990. Projekt nosi kryptonim „Radiostron" i już dziś wzbudza zainteresowanie uczonych całego świata.

— Co daje umieszczenie radioteleskopów w przestrzeni kosmicznej? — będąc w Instytucie Badań Kosmicznych w Moskwie zapytałem o to prof. Nikołaja Kardaszewa, członka-korespondenta Akademii Nauk ZSRR, czołowego specjalistę na świecie w zakresie poszukiwań pozaziemskich cywilizacji.

— Chodzi nie tylko o zwiększenie odległości między antenami, czyli o zwiększenie rozdzielczości przyrządu — odpowiada prof. Kardaszew. — Równie istotne jest całkowite wykluczenie wszystkich zakłóceń, które są wywoływane na Ziemi przez liczne radiostacje i urządzenia przemysłowe.

— W jaki punkt nieba proponuje pan profesor skierować taki radioteleskop w pierwszej kolejności?

— Pierwszeństwo należy się jądrom galaktyk i pulsarom oraz okolicom hipotetycznych „czarnych dziur" o bardzo dużych masach, rzędu miliardów mas Słońca. Interferometr ten umożliwi też wykonanie kolejnego znaczącego kroku w poszukiwaniu pozaziemskich cywilizacji.

Waldemar Siwiński

OCZY ZIEMI

„Sondy kosmiczne, obsadzenie Księżyca stacjami naukowymi i budowa nowych ziemskich obserwatoriów, a zwłaszcza większych radioteleskopów są ważniejsze od wszystkich innych budowli i konstrukcji, jakie [illegible] wznosi...

Hoimar von Ditfurth „Dzieci Wszechświata”

***Cóż** piękniejszego nad niebo, które ogarnia wszystko, co piękne — te słowa Mikołaja Kopernika zapisane w sławnym dziele „De Revolutionibus...” najlepiej wyrażają odwieczną tęsknotę człowieka do poznania tego wszystkiego co jest ponad nami, ponad Ziemią. Podstawowym — dziś i w przyszłości — instrumentem tego poznania są radioteleskopy, wspaniałe oczy i uszy Ziemi, których stworzenie możliwe było dzięki elektronice, komputerom i geniuszowi matematyków.*

Już dzisiaj radioteleskopy gruntownie zmieniły nasze wyobrażenia o kształcie Wszechświata. A jest to dopiero przedsmak tego, co nas czeka w roku 1990, gdy urzeczywistniony zostanie radziecki projekt „Radiostron”, mający na celu zbudowanie superradioteleskopu kosmicznego.
Ale po kolei.

DRUGA REWOLUCJA

Przygoda z radioteleskopami zaczęła się niespełna 55 lat temu, gdy inżynier Karl Jansky, młody pracownik Laboratorium Bella dostał polecenie zbadania zakłóceń występujących w przekazywaniu sygnałów radiowych na duże odległości. Po kilku miesiącach pomiarów okazało się, że część szumów zakłócających łączność pochodzi z „nieba”. Odkrycie to stało się na kilka lat sensacją dla dziennikarzy i szerokiej publiczności, ale fachowi astronomowie zlekceważyli je. Zignorowała je również firma Bella. Jansky zmarł w wieku 45 lat, nie zdając sobie sprawy z tego, że zapoczątkował nową dziedzinę nauki — radioastronomię. Pierwszej rewolucji w astronomii dokonali Kopernik i Galileusz. Drugą zapoczątkował Jansky. Rzecz w tym, że technika obserwacyjna zapoczątkowała lunetą Galileusza wyczerpała w pierwszej połowie XX wieku swe możliwości. Coraz trudniejsze stawało się bowiem zwiększanie rozdzielności (a co za tym idzie zasięgu obserwacji) poprzez budowę wielkich teleskopów.

Powstania radioastronomii przełamało tę barierę. Zdolność rozdzielcza i czułość obserwacji radiowych są obecnie 10 tysięcy razy większe od optycznych. Znaczy to, że obserwuje się promieniowanie radiowe od źródeł, których nie można zidentyfikować optycznie.

Pierwszą radiową mapę nieba opublikował w 1940 roku amerykański radioamator — Grote Reber. Dla astronomów „skazanych” do tego czasu na obserwacje światła widzialnego, otwarło się następne okno w kosmos.

Pierwszy profesjonalny radioteleskop wyposażony w antenę paraboliczną o średnicy 76 m powstał w Jodrell Bank pod Manchesterem. Z kolei w Cambridge zbudowano na odcinku 5 km 8 małych anten połączonych w system. Amerykanie zbudowali nieruchomą antenę o średnicy 300 metrów na wyspie Porto Rico. Radzieccy astronomowie m.in. uruchomili na Kaukazie antenę pierścieniową „Ratan 600”, której obwód wynosi 1,6 km. Dotychczas największy sterowany radioteleskop zbudowany został w RFN; naukowcy z Instytutu Maxa Plancka w Effelsburgu od 1971 roku „słuchają” nieba swoim stumetrowym uchem, wychwytując nawet najsłabsze sygnały.

Do ekskluzywnego klubu krajów mogących pochwalić się posiadaniem własnego radioteleskopu należy również Polska. Znajduje się on w Piwnicach 13 km od Torunia, na terenie Zakładu Radioastronomii Uniwersytetu im. Mikołaja Kopernika. Zachęcam, aby planując szkolne wycieczki do Torunia przewidzieć w ich planie wizytę w tym nowoczesnym ośrodku naukowym, w którym polscy radioastronomowie podpatrują życie Wszechświata. Warto!

Konsekwencją drugiej rewolucji w astronomii było radykalne przemeblowanie modelu Wszechświata. Uznawany obecnie za najbardziej odpowiadający obserwacjom „model standardowy”, oparty na teorii Wielkiego Wybuchu, w dalszym ciągu inspiruje jednak więcej pytań niż udziela odpowiedzi. I dlatego, aby na te pytania odpowiedzieć, niezbędne jest dalsze wzbogacanie bazy empirycznej kosmologii.

SUPERSYNTEZA

Jest intuicyjnie zrozumiała, że im większą dysponujemy anteną tym słabsze sygnały można odbierać. A możliwość odbioru słabszych sygnałów to zwiększenie zasięgu i rozdzielczości radioteleskopu. Dlatego właśnie od dawna starano się budować coraz większe przyrządy. Toruńska 15-metrowa antena paraboliczna wygląda z bliska okazale i imponująco, ale zestawiona na przykład ze stumetrową czaszą radioteleskopu Instytutu Maxa Plancka maleje do wymiarów karzełka.

Budowa coraz większych anten napotyka jednak na barierę rosnących wykładniczo kosztów i trudności technicznych. Wydawało się, w pewnym momencie, że, tak jak wcześniej astro-

dokończenie na str. 31